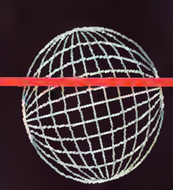
SEPTIEMBRE 1985 Nº 6 ₳ 1,50 REP. ARGENTINA





COMPUTACION PARA TODOS

21 Programas Inéditos



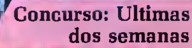
Desarrollos:

Interfases Para

TS 2068 y Commodore

Aplicaciones Comerciales,

Educativas. Utilitarios



Concurso: Ultimas dos semanas

TK-85 el microcomputador que Ud. puede usar

EL MICROCOMPUTADOR QUE LE BRINDA LAS MAXIMAS POSIBILIDADES DE LA COMPUTACION



• TK85, fue especialmente diseñado y construido para que Ud. lo use con extrema simplicidad.

• Sólo basta consultar su ameno y completo manual de instrucciones, en Castellano por supuesto, y Ud. podrá del computación en forma fácil, rápida y práctica y en lenguaje BASIC.

• A partir de allí Ud. podrá preparar sus propios programas, o también utilizar centenares de programas que existen en el mercado para estudiar: matemáticas, física, química, biología; música, catalogar clientes; controlar stocks; programar compras y ventas; controlar su cuenta bancaria, poner en orden gastos e impuestos en su hogar, jugar ajedrez; backgamond; y ¿por qué no? con su exclusivo joystick disfrutar de los más fantásticos video juegos: guerra de las galaxias, invasores espaciales; etc. y todo lo que Ud. puede imaginar.

El único límite del microcomputador MICRODIGITAL TK85 es "SU IMAGINACION".

OPCIONALES:

• Joystick (órgano de comando externo), para juegos de video • Generador de sonido • Expansor de memoria de 64 Kbytes RAM • Impresora directa • Programador de EPROM • Interface para conexión de impresora profesional (paralela).



MICRODIGITAL TK-85

Importa y distribuye. ARVOC s.a.i.c.f.i.
Avda. DIAZ VELEZ 4147 (1200) Capital Tel.: 981-1980/9212

GARANTIA: 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electrónica, fotografía y librerías

S U M A R I O

TS 2068: INTERFASE PARA IMPRESORA

Para conectar a la TS 2068 una impresora que utiliza papel común, desarrollamos una interfase de hardware.

En pág **28**

PROGRAMAS INEDITOS

TS 1000/1500, CZ 1000/1500, TK 83/85
- Números Mágicos (pág. 46)
- La invasión de los seteros (pág. 48)
- Ataque espacial (pág. 8)
- Estrellas (pág. 9)
- Invasión de letras (pág. 8)
- Gool (pág. 8)
- Sistema de ecuaciones (pág. 9)
- Cavernas (pág. 50)
- Sistemas (pág. 50)
- Agenda (pág. 10)

Spectrum, TS 2068 y TK 90X
- Rótulo (pág. 19)
- Tarántula (pág. 20)
- Batalla Naval (pág. 52)
- Musico (pág. 18)

TI 99/4A
- Ranitas (pág. 32)
- Generadoras de caracteres (pág. 34)
- Joysticks (pág. 33)

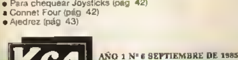
Commodore 64
- Directorio (pág. 41)
- Para chequear Joysticks (pág. 42)
- Connet Four (pág. 42)
- Ajedrez (pág. 43)



CARTA DEL DIRECTOR

Unos cien mil estudiantes del nivel primario utilizan computadoras en nuestro país, cifra que coloca a la Argentina entre las naciones más avanzadas en esta materia, según advirtió el doctor Antonio Battro durante las Jornadas sobre Educación y Computación auspiciadas por K 64 (de las que informamos en esta edición). Señaló el experto que nuestra actividad en informática es excepcional, afirmación que a través de K 64 podemos corroborar, particularmente por la cantidad y la calidad de los programas que recibimos en nuestro segundo gran concurso. Respondemos a este fenómeno aumentando la calidad del software y de las notas que publicamos, **incrementando nuevamente el número de páginas (sin alterar el precio)** y lanzando al mercado una tirada aun mayor. Pedimos disculpas a quienes no pudieron conseguir el número cinco, ya que en muchos lugares se agotó. Y agradecemos a todos los lectores, que hicieron posible este éxito editorial.

CRISTIAN PUSSO



COMO USAR LA INTERFASE SERIE DE LA C64

La máquina tiene incorporada una interfase serie que se puede usar fácilmente desde el Basic para enviar y recibir datos desde otra computadora o transmitir a una impresora serie.

En pág. **38**

OTRAS NOTAS

- Desentrañando los códigos (Peek-Poke) (pág. 14)
- Alta resolución en la CZ 1000 (pág. 24)
- El futuro de Sir Clive Sinclair (pág. 12)

K64 COMPUTACION PARA TODOS

AÑO 1 Nº 6 SEPTIEMBRE DE 1985

Director General
Ernesto del Castillo
Director Editorial
Cristian Pusso
Director Periodístico
Fernando Flores
Director Financiero
Javier Campos Malbrán
Coordinador
M.G. Verdomer Weiss
Secretaria
Moni Ocampo
Departamento de Publicidad
Jefe: Dolores Unen
Promotora
Mónica Garibaldi

Departamento de Avisos
Oscar Devoto
Diagramación y Armado
Fernando Amengual
Servicio de fotografía
Juan José Perez
Esteban Figueredo

Foto de tapa: Gráficos artísticos generados por computadores

K-64 es una Revista mensual editada por Editorial PROEDI S.A. [illegible] Carrito 1320 1º Piso Buenos Aires Te. 42-9661/8 Registro Nacional de la Propiedad Intelectual 313 837 M registrada. Queda hecho el depósito que indica la Ley 11.723 de Propiedad Intelectual. Todos los derechos reservados.
Precio de este ejemplar un austral con cincuenta centavos. Precio de la suscripción semestral 8 australes.
Distribuidor en Capital: Infieria, Venezuela 1417 Capital Federal Tel. 37-6664.
Impresión Calcotam. Fotocromía tapa Columbia. Fotocomposición Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

ATARI con nuevos propietarios.

Hace ya algunos meses se comentaba la posible venta de ATARI. La Werner Comunications, propietaria de hasta ese momento de la popular marca de video-juegos y ordenadores, se había manifestado incapaz de gestionar con buen término económico la



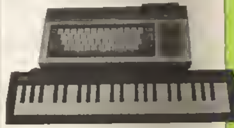
transición de ATARI del mundo de los video-juegos al mundo de los ordenadores.
El comprador con mayores posibilidades parecía ser Philips, gigante de la electrónica europea. No obstante, la venta terminó por realizarse a Jack Traimel, 20 millones de dólares de por medio.
Traimel empezó a poner en práctica sus conocimientos aprendidos al colaborar en la creación de Commodore International, cuya repercusión en el campo de las computadoras del hogar es sustantiva.
Traimel adquirió la marca e infraestructura de fabricación y distribución de ATARI. Aplicando su experiencia de construir a bajo costo, lanzó hace poco tiempo los computadores ATARI 600 XL y 800 XL, que llevaron nuevamente a la firma al liderazgo mundial.
Pero Traimel desea llegar aún más lejos con su nueva empresa.
Al respecto se espera que pronto aterrice en el mercado un nuevo equipo capaz de competir con el Macintosh de APPLE y el PC Junior de IBM, no sólo en prestaciones, sino con un precio bastante más bajo que la mitad de los arriba nombrados.

Computadora de pulsera.

No hay duda que la micro-electrónica es un terreno abonado para los japoneses. En esta ocasión, la novedad viene de la mano de SEIKO, adoptándo la forma de un tradicional reloj de pulsera. El RC-1000 es un terminal de pulsera que puede recibir datos directamente de un ordenador personal. Posee una capacidad de 2 K bytes y la transferencia de información procedente del ordenador se realiza en un tiempo máximo de 10 segundos. En este intervalo de tiempo, la información seleccionada puede transferirse al RC-1000, sin otro requisito que conectarle una entrada RS-232 standard. Dada su capacidad de memoria, se pueden almacenar en él hasta 80 "páginas" de 24 caracteres.

YIS-503 o la mezcla musico-informática.

Los japoneses, han lanzado recientemente al mercado el YIS-503, que no es un equipo con alguna que otra posibilidad sonora, ¡es un músico por vocación! Su diseño se adscribe al novedoso standard MSX, que establece una normalización básica tanto en software como en hardware, de tal forma que cualquier ordenador acogido a esta filosofía, es compatible con el resto de la familia MSX.
La firma YAMAHA es la creadora del invento, bautizado como YIS-503.
Con un cerebro basado en el microprocesador Z-80 A, incorpora en su versión base 32 Kbytes de memoria ROM y una RAM de 48 Kbytes. A través de posteriores ampliaciones, el equipo puede llegar a direccionar la friolera de 1 Mbyte, repartidos en 16 bloques de 64 Kbytes.
El teclado tipo QWERTY consta de 63 teclas, incluidas las de desplazamiento del cursor, más otras cinco de control. Posee además la posibilidad trabajar con cartuchos en ROM.
La presentación en pantalla en modo texto adopta un distribución de 24 líneas por 32 o 40 columnas, mientras que la resolución gráfica es de 256 x 192 pixels.
Pero tal vez lo más saliente sea su facilidad para hacer música, dispone de ocho octavas y tres notas por acorde con control de envolvente. El genio que orquesta el conjunto sonoro es el sintetizador SFG-



01 FM, que por si fuera poco, permite la elección de 48 voces distintas, guitarra eléctrica y clásica, clarinete, piano,... además de generar una ingente variedad de sonidos: sirenas, sonido de lluvia, etc.
La gama de ritmos tampoco es escasa: jazz rock... Y una opción interesante: permite acoplarse al equipo un teclado de tipo piano como si se tratara de un periférico más. Por supuesto, la composición musical se va registrando en forma automática en la memoria RAM, para su posterior grabación en disco, cassette, etc.
El inconveniente radica en el uso obligado del receptor de T.V. aunque ya se están pensando etapas de expansión con amplificadores y equipos de sonido de mayor potencia.
Verdaderamente cada día nos vemos más asombrados con los nuevos logros de las computadoras, que

parecen arremeter contra cualquier tipo de información como en este caso, musical

Novedades para los Spectrumnomanos:

Una de las pocas cosas que no podíamos hacer con nuestra Spectrum era hablar con ella. Bien recientemente se ha lanzado en el mercado europeo un tipo de accesorio que nos permite hacerlo.
Se trata ni más ni menos de un sintetizador de voces alófonas.
Algunos se preguntarán ¿qué es eso? Simplemente, utilizar sonidos vocales individuales que, una vez reunidos, forman voces inteligibles.
Son tres las marcas que publicitan sus respectivos generadores de voz. Pero hay una en particular, CURRAH, que la interface que presenta posee características no igualadas por sus competidores directos. Estas son: no utilizar un vocabulario fijo, por lo que es capaz de sintetizar cualquier palabra o sentencia, el sonido se mezcla con la salida de video, efecto que hace que escuchemos a través del parlante de nuestro televisor.
Los sonidos se introducen a través de cadenas. Hay que asignar a la variable S$ valores determinados. Por ejemplo:

LET S$ = "hola"

Al ejecutar esto escucharemos la palabra "hola".
El único punto en contra, para nosotros, es que los alófonos son dominantemente anglosajones; por lo que hay que dominar muy bien las ordenes para "sacar" palabras claras.
En síntesis, es un "chiche" muy divertido, y sería muy grato tenerlo entre nosotros, ya que con un "aparato" de este tipo podrían agregarse infinidad de efectos a nuestros programas.

MSX en la Argentina

Uno de los problemas de la industria informática es el de la "incompatibilidad" de equipos y sistemas. Como solución se ha planteado la "standerización". En esto están ASCII, una exitosa editorial productora y distribuidora de software de Japón y la Microsoft Corporation. Crearon un standard, en colaboración con los principales fabricantes japoneses, con el deseo de que se la reconociera internacionalmente como tal. El resultado final de esta tarea de investigación y desarrollo se denominó Norma MSX.
Las especificaciones del producto incluyen exigencias basicas de Hardware, basadas alrededor de un microprocesador Z-80 y otros chips fijos, como así también un lenguaje standarizado.
El concepto MSX consiste en potenciar las innovaciones tecnológicas a través de especificaciones básicas para el Hardware y Firmware sobre las cuales cada fabricante puede concretar su producto.
Como factica común en el mercado de computadores hogareños, importantes empresas adquirieron la licencia MSX.
En la Argentina la implementa Telent, con su nueva



planta industrial en la provincia de San Luis.
Cada fabricante está diseñando su modelo propio con características ampliadas que pueden ser aprovechadas por el resto de los usuarios. No sólo se pueden apreciar las ventajas a nivel de desarrollos en Hardware, sino también a nivel de Software, ya que no se trabaja para una marca en particular, sino para un standard universal.



K-64 en Expoficina



Estuvimos en Expoficina '85. Allí recibimos la visita del intendente doctor Julio César Saguier (en la foto a la derecha) acompañado por nuestro director editorial, Cristian Pusso (izquierda).
En el próximo número daremos amplia información sobre esta muestra dedicada a la informática

Intercambio

La Asociación Argentina para Desarrollo de la Tecnología Educativa organiza las Segundas Jornadas de Intercambio de Experiencias de Informática Educativa, con sede en la Escuela Argentina Modelo de Capital Federal, para el próximo 29, 30 y 31 de Octubre.
El propósito de la reunión es posibilitar la difusión de las experiencias pedagógicas desarrolladas por los establecimientos escolares que afrontaron el empleo de la computación como elemento auxiliar de la educación. El análisis de la amplia gama de posibilidades que brinda el computador y la promoción de los avances obtenidos, podrán ser apreciados en las exposiciones, las mesas redondas y la feria de computación que se realizará el 31 de octubre.
Los interesados pueden dirigirse a la secretaría de la Asociación, te: 33-0610 / 33-0252, Perú 277 1º Piso, Capital Federal, o a la secretaría de la Escuela Argentina Modelo, Riobamba 1059, Te. 41-3266.

Educación y Computación

Se realizaron las Primeras Jornadas de Educación y Computación para Docentes y el Primer Centro de Información y Muestra de Elementos de Computación. Tuvieron lugar en el Auditorio Buenos Aires, organizadas por Epi y Sanwa, con el auspicio de K-64, Aguilar, Czerweny y Radio Victoria y el apoyo de la Secretaría de Educación de la Municipalidad de Buenos Aires
Las jornadas fueron dirigidas a educadores, pedagogos y autoridades educativas oficiales y privadas. Disertaron sobre temas de interés los doctores: Alicia Bañuelos, Antonio Battro, Máximo Salvi; ingenieros Ana María Velo, Gustavo Pollitzer, Sergio Agoff, Aníbal Mellano y Héctor Thompson; licenciados Adolfo D'Onofrio y Eros Galván.
La respuesta por parte del público concurrente hizo de estos eventos un éxito, evidenciando el peso que la computación ocupa en el ámbito de la educación.

Arte Infantil

La galería de arte Ruth Benzacar, de Buenos Aires, abrió sus puertas a la Asociación Amigos del Logo, para realizar una muestra de "Arte Infantil y computadoras".



Doctor Battro de la Asociación Amigos de Logo y Ruth Benzacar acompañando a los niños

Participaron artistas muy jóvenes, todos ellos alumnos de escuelas primarias de la Capital Federal y de las provincias
Todos los niños que visitaron la Galería fueron invitados a crear nuevas obras "fuera de catálogo" con

INVASION DE LETRAS

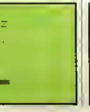
COMP: CZ-1000/TK-83/85
CONF: 2 K
CLAS: ENT



Este programa comanda una invasión de letras. Para repelerla debemos responder con la letra que se muestra en pantalla. Para terminar y ver el puntaje hay que pulsar la tecla "1".

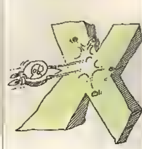
```
  1 REM [illegible]
  2 REM [illegible]
  3 REM *INVASION DE LETRAS*
  4 REM [illegible]
  5 LET H=0
 10 CLS
 15 LET Z=-1
 20 LET K=9
 25 PRINT AT 10,K
 30 LET L=INT (RND*26)+38
 35 FOR F=0 TO 18
 40 PRINT AT 3 F  ;CHR$ L
 45 IF INKEY$=CHR$ L THEN GOTO
55
 50 IF INKEY$="1" THEN GOTO 145
 55 NEXT F
 60 GOTO 10
 65 IF F>K THEN GOTO 90
 70 FOR N=K TO F+1 STEP -1
 75 PRINT AT 10 N
 80 NEXT N
 85 GOTO 105
 90 FOR N=K TO F-1
 95 PRINT AT 10 N
100 NEXT N
105 IF F=1 THEN LET Z=-Z
110 FOR F=S TO 2 STEP -1
115 PRINT AT F+1,N-2      AT F N
-1
120 NEXT F
130 PAUSE 25
135 LET H=H+1
140 GOTO 10
145 CLS
150 PRINT "SCORE= " H
160 STOP
170 SAVE "INV LETRAS"
180 RUN
```

Pantalla

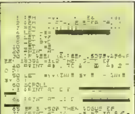


ATAQUE ESPACIAL

COMP. CZ-1000/TK-83/85
CONF. 2 K
CLAS: ENT



Con nuestro vehículo espacial ("H"), que debemos conducir con las teclas "5" y "8", destruiremos a las "x" enemigas, pasándoles por encima.



GOOL

COMP CZ-1000/TK-83/85
CONF 2 K
CLAS: ENT



Este pequeño programa tratará de "hacernos goles". Deberemos evitar que el marcador se incremente. Para ello podemos controlar nuestro "arquero" con las teclas "6" y "7". Finalizadas las jugadas, la computadora indicará nuestro puntaje.

```
  1 REM [illegible]
  2 REM [illegible] GOOL [illegible]
 10 LET [illegible]=0
 20 LET [illegible]
 30 FOR [illegible]
 40 LET [illegible]=INT [illegible]
 50 FOR [illegible]
 60 PRINT AT [illegible]
[illegible]
 70 LET [illegible] INKEY$ [illegible]
[illegible] INKEY$ [illegible]
 80 IF C= [illegible]
[illegible]
 90 IF [illegible] THEN LET S=S-1
100 NEXT C
110 IF [illegible] GOTO [illegible]
120 PRINT AT [illegible] "GOOL"
130 IF [illegible] THEN LET [illegible]
140 GOTO [illegible]
150 LET [illegible]
160 PRINT AT [illegible]
170 NEXT [illegible]
180 PRINT AT [illegible]
[illegible]
```

Pantalla

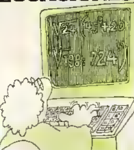
GOOL

SISTEMA DE ECUACIONES



COMP: CZ-1000/TK-83/85
CONF: 16 K
CLAS: EDU

Publicamos una nueva versión de un programa para resolver sistemas de ecuaciones.
Las instrucciones son muy simples y aparecen en pantalla.
Esperamos que les resulte de utilidad.



Pantalla

[illegible]

```
[illegible]
```

ESTRELLAS

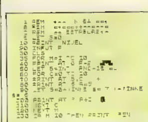
COMP: CZ-1000/TK-83/85
CONF. 2 K
CLAS: ENT



Nuestro objetivo es guiar una nave de manera que quede en posición de bombardear la mayor cantidad de estrellas posibles.
El programa finaliza si derribamos 10 estrellas.

```
[illegible]
```

Pantalla



AGENDA

COMP: CZ-1000/TK-83/85
CONF: 16 K
CLAS: PERS.

Este programa puede convertirse en una herramienta muy útil, para todos aquellos que tengan largas listas de actividades. También servirá de "apuntador" para poder individualizar lugares, fechas, y otros datos.

Es una agenda tipo, o más bien, un pequeño libro de notas. La utilización es en base a respuestas realizadas por el programa, donde deben seleccionarse las opciones adecuadas, salvo en aquellas ocasiones en que debe entrarse un código particular para salir o regresar al menú. Estas situaciones se dan en general cuando por ejemplo estamos cargando nuestras notas y queremos indicar que terminamos; para ello entrar "ENTER". Los códigos aceptados como respuestas son, según el caso; teclear un punto como respuesta; teclear un cero como respuesta; teclear "ENTER" como respuesta.

Su uso es muy simple. Con sólo unos minutos de "manejar" este programa, llegamos a dominarlo por completo.

Tal vez resulte un tanto vacío y escaso de pantallas y explicaciones. La razón es clara: cuanto menos memoria ocupe el programa, más lugar para nuestras notas quedará. Por esto es que el estilo es bastante "espartano".

Con respecto al programa en sí, es un archivo que utiliza strings (cadenas) para entrada y ordenamiento: B$ y C$ para entrada que son reubicados en la rutina de ordenamiento a partir de la línea 505. Por lo demás, el resto del programa es fácilmente "deducible".

Esperamos que este trabajo aclare algunas dudas que han planteado lectores con respecto a este tipo de programas, su ensamble y funcionamiento.

```
10 DIM A$ [illegible]
20 DIM B$ [illegible]
30 DIM C$ [illegible]
50 PRINT "LIBRO DE NOTAS"
[illegible]
65 INPUT N
60 IF N<1 OR N>6 THEN GOTO 55
65 GOTO N*100+(N-1)*100
100 GOSUB 1500
105 FOR I=1 TO 120
110 IF A$(I,1 TO 1)=" " THEN GOTO 140
115 NEXT I
120 PRINT AT 20,0;"NO QUEDA MAS ESPACIO"
125 IF INKEY$="" THEN GOTO 125
130 CLS
135 GOTO 50
140 PRINT AT 20,0;"FECHA DD"
145 INPUT D
147 IF D=0 THEN GOTO 130
150 IF D<1 OR D>31 THEN GOTO [illegible]
155 PRINT "MM"
160 INPUT M
165 IF M<1 OR M>12 THEN GOTO 160
170 IF M=2 AND D>29 THEN GOTO [illegible]
171 GOTO 180
172 GOSUB 1500
175 GOTO 140
180 PRINT "AA"
185 INPUT A
190 IF A<100 THEN GOTO 185
[illegible]
250 PRINT AT 20,0;"INTRODUZCA LA PRIMERA LINEA DE LA ANOTACION"
255 INPUT B$(I,1 TO 32)
260 PRINT AT 20,0;"INTRODUZCA LA SEGUNDA LINEA DE"
[illegible]
300 GOSUB 1500
305 PRINT AT 20,0;"FECHA DE BUSQUEDA AAMMDD"
310 INPUT V$
311 IF V$="" THEN GOTO 330
312 IF LEN V$<>2 OR LEN V$>6 THEN GOTO 310
315 FOR I=1 TO 120
320 IF A$(I,1 TO LEN V$)=V$ THEN GOTO 350
325 NEXT I
330 GOSUB 1500
335 PRINT AT 20,0;"NO HALLADO"
337 IF INKEY$="" THEN GOTO 337
340 GOTO 330
350 GOSUB 1500
355 PRINT AT 8,2;A$(I,5 TO 6);[illegible]
360 PRINT AT [illegible];B$(I,1 TO 32);AT 9,2;C$(I,1 TO 32)
365 IF INKEY$="" THEN GOTO 365
368 GOSUB 1500
370 GOTO 325
400 CLS
495 GOTO 50
500 GOSUB 1500
505 PRINT "[illegible] VOY A TRABAJAR EN MODO FAST. SU PANTALLA PERMANECERA [illegible] HASTA EL FIN DE LA CLASIFICACION"
510 IF INKEY$="" THEN GOTO 510
515 LET [illegible]
520 LET [illegible]
525 LET [illegible]
530 FAST
535 FOR P=119 TO 1 STEP -1
540 FOR Q=1 TO P
545 IF A$(Q,1 TO 6)<=A$(Q+1,1 TO 6) OR A$(Q+1,1 TO 1)=" " THEN GOTO 600
550 LET [illegible](1 TO 6)=A$(Q,1 TO 6)
555 LET U$(1 TO 32)=B$(Q,1 TO 32)
560 LET X$(1 TO 32)=C$(Q,1 TO 32)
565 LET A$(Q,1 TO 6)=A$(Q+1,1 TO 6)
570 LET B$(Q,1 TO 32)=B$(Q+1,1 TO 32)
575 LET C$(Q,1 TO 32)=C$(Q+1,1 TO 32)
580 LET A$(Q+1,1 TO 6)=V$(1 TO 6)
585 LET B$(Q+1,1 TO 32)=U$(1 TO 32)
590 LET C$(Q+1,1 TO 32)=X$(1 TO 32)
600 NEXT Q
625 NEXT P
630 SLOW
635 GOSUB 1500
640 PRINT AT 11,0;"CLASIFICACION COMPLETE"
645 IF INKEY$="" THEN GOTO 645
650 CLS
655 GOTO 50
700 GOSUB 1500
705 PRINT "1-PANTALLA 2-IMPRESORA"
710 INPUT L
715 IF L<1 OR L>2 THEN GOTO 710
720 IF L=2 THEN GOTO 800
725 FOR K=1 TO 120 STEP 3
730 GOSUB 1500
735 PRINT
736 FOR J=0 TO 2
737 IF A$(J+K,1 TO 1)=" " THEN GOTO 745
740 PRINT A$(J+K,5 TO 6);[illegible];A$(J+K,3 TO 4);[illegible];A$(J+K,1 TO 2);B$(J+K,1 TO 32);C$(J+K,1 TO 32)
742 PRINT
745 NEXT J
750 LET V$=INKEY$
755 IF V$="" THEN GOTO 750
760 IF V$="." THEN GOTO 770
765 NEXT K
770 CLS
775 GOTO 50
800 FOR K=1 TO 120
825 LPRINT A$(K,5 TO 6);[illegible];A$(K,3 TO 4);[illegible];A$(K,1 TO 2)
830 LPRINT
835 LPRINT B$(K,1 TO 32)
840 LPRINT C$(K,1 TO 32)
850 LPRINT
855 LPRINT
860 NEXT K
870 CLS
875 GOTO 50
900 GOSUB 1500
905 PRINT "1-TOTAL 2-PARCIAL"
910 INPUT L
915 IF L<1 OR L>2 THEN GOTO 910
920 IF L=2 THEN GOTO 980
925 FOR J=1 TO 120
935 PRINT AT 11,12;"BORRANDO"
940 LET A$(J,1 TO 6)=[illegible]
945 LET B$(J,1 TO 32)=[illegible]
950 LET C$(J,1 TO 32)=[illegible]
955 NEXT J
960 PRINT AT 11,5;"BORRADO COMPLETO"
965 IF INKEY$="" THEN GOTO 965
970 CLS
975 GOTO 50
980 PRINT AT 20,0;"DESDE LINEA"
985 INPUT H
1000 IF H<1 OR H>120 THEN GOTO [illegible]
1005 PRINT AT 20,0;"HASTA LINEA"
1010 INPUT G
1015 IF G<H OR G>120 THEN GOTO 1010
1017 GOSUB 1500
1020 FOR J=H TO G
1025 PRINT AT 11,12;"BORRANDO"
1030 LET A$(J,1 TO 6)=[illegible]
1040 LET B$(J,1 TO 32)=[illegible]
1045 LET C$(J,1 TO 32)=[illegible]
1050 NEXT J
1055 GOTO 960
1100 GOSUB 1500
1110 PRINT "PREPARE EL CASSETTE PARA GRABAR EL LIBRO DE NOTAS"
1120 PRINT "[illegible] PULSE NEW-LINE"
1130 INPUT [illegible]
1140 SAVE "LIBRO DE NOTAS"
1150 CLS
1160 GOTO 50
1500 CLS
1505 PRINT "LIBRO DE NOTAS"
1510 GOTO 1520
1520 PRINT "1-[illegible]"
1521 RETURN
1522 PRINT "2-BUSQUEDA"
1523 RETURN
1524 PRINT "3-CLASIFICACION"
1525 RETURN
1526 PRINT "4-IMPRESION"
1527 RETURN
1530 PRINT "5-BORRADO"
1531 RETURN
1532 PRINT "6-GRABACION"
1533 RETURN
1600 PRINT AT 20,0
1605 RETURN
1650 IF LEN V$=2 THEN RETURN
1655 LET V$="0"+V$
1660 RETURN
```

TV-TEST

PROGRAMA	ME GUSTA MUCHO	POCO	NADA
Cuentos para Ver Los cuentos más interesantes de nuestra literatura, hechos especialmente para televisión.			X
Cable a Tierra Un espacio libre, para que los jóvenes se expresen			X
La Memoria Una miniserie conmovedora dirigida e interpretada por los grandes de la escena nacional			X
Colorín Colorado Un teleteatro que renueva la televisión, hecho por gente muy joven			X
El País que debemos cambiar La polémica necesaria, entre el país que somos y el que debemos ser			X
Encuentros Los grandes de la historia superan épocas y geografías, para tener un encuentro divertido e inteligente			X
Chicas y Chicos Un programa para que los chicos se diviertan en serio y para que los padres los entiendan mejor		X	X
Cine Club Las mejores películas de la historia del cine, en su idioma original			X
Argentina Secreta La aventura fascinante de descubrir nuestro país			
Nos estamos viendo Dos especialistas critican con toda libertad la televisión y nuestra propia televisión.			X

Estos programas son algunos ejemplos de la televisión que queremos hacer. Y necesitamos su opinión

Por favor vea estos programas, luego marque con una (x) en el casillero correspondiente, arranque la hoja y envíela a: TV-TEST, ATC, Av. Figueroa Alcorta 2977 (1425) Bs. As.

NOMBRE ____________

DIRECCION ____________

LOCALIDAD ____________

TELEFONO ____________

ATC

En todo el país

El futuro de Sir Clive Sinclair

Allá por el año 1980, sir Clive Sinclair cansado de experimentar en el campo de la microelectrónica con sus calculadoras científicas y programables de notación polaca, decidió iniciar un proyecto que probablemente en ese tiempo, no tenía ni idea del revuelo informático que en un escaso futuro iba a producir.
Todo comenzó con la loca idea de construir una computadora que sea lo suficientemente barata como para que cualquier vecino de barrio pueda darse el lujo de programar. Pero aquí no termina la cosa, este nuevo "micro" debía además estar al alcance no sólo de los bolsillos de cualquiera sino, también, del de un cerebro tipo medio sin "chispazos" de genio y que dispusiera de un rato de ocio, un grabador, un televisor, un manual y un poco de ganas para dilucidar el hasta entonces "tétrico" mundo de la informática. Bajo estas premisas nació a mediados del '80 el primer "fierro" de sir Clive: la ZX-80.
Hoy la podemos catalogar como "precaria", indicada para principientes eternos, pero la verdad es que esta maquinita fue el comienzo del revuelo. Estaba equipada con una ROM de 4 K mas 1 K de memoria RAM expandible a 16 K y el microprocesador era el tan mentado Z-80 A a 3.25 MHz, por supuesto de ocho bits. La pantalla era de 24 líneas por 32 caracteres y la velocidad de transmisión de datos a cassettes era de 250 baudios, todo un mounstro.
Pero el ZX-80 era el principio de un desarrollo que en su etapa siguiente derivó, allá por el '81, en el archifamoso y supervendido en todo el mundo ZX-81. Esta afirmación la prueban las ventas de 2 millones de máquinas los dos primeros años, que fueron a engrosar de buena gana, las arcas de sir Clive. La producción estaba a cargo de la firma TIMEX de Escocia. Semejante "rating" catapultó al ZX-81 a muchos países, a través de licencias de representación y ventas.
Los equipos son ensamblados en Portugal, siguiendo el pie de la letra, las instrucciones de don Sinclair, de mandar a hacer todo en otra parte, y Sinclair Research, se reserva el trabajo de diseño y proyecto de nuevos equipos, la producción y todo eso, afuera



El ZX-81 original venía equipado con 8 K de ROM, 1 K de RAM expandible a 16 K, con el Z-80 también a 3.25 MHz. Esta primera versión en sus primeros modelos, tenía "pinchada" la ROM, así es, con el uso le descubrieron alguno que otro "bug" que obviamente se les escapó al grupo de matemáticos de Cambridge que la diseñó.
Este "bug" se corrigió enseguida y a partir de allí no sucedieron nuevos errores. A continuación, la versión que se ensambló en Portugal, y que llegó a nosotros, tenía 2 K de RAM. El éxito alcanzado no solo agrandó la cuenta de sir Clive, sino también la de muchísimos fabricantes de accesorios, periféricos y de interfaces, (algunos muy interesantes y otros realmente dignos del tachito de la basura). Y ni qué decir de los creadores de software. Inventaron soft que hacía que la máquina casi pudiera pensar sola, y eso que la velocidad de operación es bastante, lenta y las posibilidades gráficas bastante reducidas (ni qué hablar además del blanco y negro).
Bajo este concepto llegamos a comienzos de 1984, cuando Sinclair Research Ltd. puso en el mercado el sistema de almacenamiento masivo que llamo Microdrives. Consiste en una interface (llamada interface 1), que se enchufa en el conector trasero. Dispone de salida RS-232, la conexión para las unidades microdrives, y conexión para armar una red (net) que da la posibilidad de conectar hasta 64 Spectrum entre sí. Los Microdrives son sistemas de almacenamiento masivo en cinta. Utilizan un pequeño cartucho, que mide aproximadamente 3 cm x 3.5 cm x .5 cm de alto y el sistema de arrastre de la cinta es del tipo de aquellos "maga-

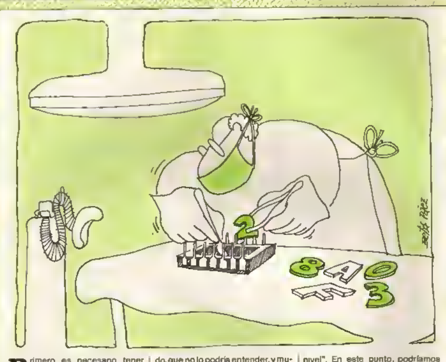
Después de lanzar las microdrives y la Interfase 2, Sinclair presentó su nuevo equipo QL, de 16 bits, que lleva 128 K bytes de RAM y anunció una tarjeta de expansión que ampliará la RAM a 640 K bytes. También hiza nacer al Spectrum + y ya se camenta la aporición del Spectrum portátil, a pilas.

zines", es decir por arrastre sin fin. En cada cartucho puede almacenarse 100 K de información, sea programas, bloques de variables, etc. Cada interface 1 puede soportar hasta 8 microdrives, por lo que la capacidad máxima es de 800 K en un solo Spectrum. Ni qué hablar si formamos una red.
Pero no todo es color rosa. Parece ser que la Interface RS-232 no es tan standar, según se comenta. El problema reside en el dispositivo para recibir datos, es decir que la trasmisión se perfecta, por lo que no acarreará problemas al usárnca una impresora. Pero si "enchufamos" un modem, la cosa se pone negra. La razón es simple. Sinclar para reducir costos sin sacrificar beneficios, optó por la solución barata de controlar todo por Software, y esto pareca complicar las cosas. Volveremos a estos temas en artículos futuros, para tratarlos con mayor espacio y tranquilidedd.
LLegamos a principios de 1982 y otra vez Don Clive es noticia; sacó casi de la galera al sucesor del ZX-81, la hoy famosa Sinclair Spectrum. Este nuevo modelo salió en dos versiones de 16 K y 48 K, ya con alta resolución gráfica, ocho colores, un "beeper" para sonido programable en 32 octavas, el viejo Z80 A ahora trabajando a 3.5 MHz, y una ROM de 16 K, en la que el BASIC de la ZX-81 quedaba totalmente inscripto, un tecladito con "teclas de verdad", reemplazando las membranas de los modelos anteriores, 21 caracteres gráficos definibles por el usuario, una velocidad de transmisión de información a cassette de 1500 baudios, en fin, una máquinita lo suficientemente "piola" como para satisfacer las ambiciones de cualquier usuario modio. Y aquí de nuevo, la avalancha de constructores de periféricos, interfaces, y qué decir del software. Con todas las mejoras técnicas, se logran programas que son verdaderas obras de ingenio e inteligencia.
A esta altura podamos comentar algo de la filosofía de sir Sinclair. Lo [illegible] sus productos es el factor económico. Es de vital importancia para él estar a "tiro" de todos los bolsillos, condición muchas veces discutida, pero de resultados evidentes.
Otra de sir Clive, fue el lanzamiento de la Interface 2, que permite adosar 2 Joysticks y con una entrada para cartuchos de programas en ROM (algo similar a los cartridges de la 2068). Pero este nuevo invento no tuvo la aceptación que se esperaba, dado que existen en el mercado interfaces de mejores prestaciones que la 2 de Sinclair. Esperemos que tanto la interface 1 como los microdrives y las interfaces para joysticks, lleguen a nuestro país para poder expandir nuestra máquina con todos estos chiches.
Por último, el 12 de enero de 1984, Sinclair lanzó al último de sus "aparitos". Se trata de un nuevo "fierro" de la fábrica de Surrey, denominado QL, por "quantum leap" (su traducción podría ser salto cuántico).
Efectivamente Don Clive está convencido de que su nuevo modelo representa un salto gigantesco en el mercado de los ordenadores. Según él, sus rivales son (y lo dice sonriente) el IBM/PC y el Apple IIe, solo que -siempre según Clive- estas máquinas se venden a precios cuatro veces superiores al del QL. Se trata de un equipo de 32 bits, basado en el microprocesador Motorola 68008. Lleva 128K bytes de RAM, y se anuncia una tarjeta de expansión que ampliará la RAM a 640 K bytes. ¿Todo un infierno, verdad?. Dos unidades Microdrives incorporadas se encargan del almacenamiento masivo.
Pero (siempre hay "peros") un aspecto riesgoso de este asunto es que el QL se rige por un sistema operativo propio, el QDOS, lo que lo hace absolutamente incompatible con cualquier máquina. Por otro lado, este aparato funciona con una versión exclusiva del BASIC al que llamaron Sinclair Super BASIC.
Los cuatro programas hasta ahora editados son: un procesador de textos, una tabla de análisis finan- [illegible] comerciales, todos de la firma Psion.
A propósito del Spectrum, el QL no es compatible. En cambio sí lo es a nivel de conexión, con las dos interfaces y el Microdrive. Un punto flojo es que no posee facilidades para conectar un Hard Disk, cosa que le podría dar una capacidad de almacenamiento impresionante.
Como es tradicional en los productos Sinclair, el QL será fabricado por una firma externa, en este caso por Datatech, una filial de Thom EMI.
No sabemos si este pequeño monstruo" pueda llegar a nuestro país, pero creamos que si gana en popularidad y eficiencia junto con un buen apoyo de software, quizá lo tendremos entre nosotros dentro de algún tiempo.
Y aquí la última de Sir Clive, este año lanzó lo que dio en llamar Spectrum +, (léase plus). Es ni más ni menos que una Spectrum, pero con una "lavadita de cara". Viene con formas cuadradas, un teclado mejorado, de plástico inyectado duro, y teclas auxiliares como por ejemplo, la coma, el punto y coma, comillas, RUN, etc. Están colocadas en teclas separadas, por lo que la velocidad y comodidad de tecleado aumenta considerablemente. El punto flojo de un teclado tan mejorado es la ausencia de Key Pad, que es ideal para el manejo de cifras, así como la carga de esas "durísimas" sentencias DATA.
Y ya como noticia fresquita, tenemos que el comentario en Inglaterra y Europa es un posible "Spectrum portátil", a pilas, con más de 100 K de RAM, pantalla incorporada de televisión plana (ya se venden en Europa); en fin, todo un chiche. Pero esta última máquina se sólo conjeturas y comentarios, al menos por ahora, y de las puertas de Sinclair para fuera, porque puertas adentro, silencio absoluto.
Bien hasta aquí hemos hecho una pequeña revisión del mundillo Sinclair, como para entender mejor las raíces de nuestro mercado actual, conociendo sus orígenes y tal vez amasgando un pronóstico hacia el futuro. De todas maneras todavía hay cuerda para volver a abordar [illegible]

Hugo Busso



DESENTRAÑANDO LOS CODIGOS



Primero es necesario tener clara la estructura interna del ZX-81, así como su funcionamiento.
Como todos sabemos, un computador tiene lo que podemos definir como "cerebro", que en el caso del ZX-81, es el microprocesador Z-80 A. Pero para dar soluciones a cada problema no basta solamente con pensar, es necesario comunicarse; y para eso, existen el teclado, a través del cual el Z-80 recibe información, y todos los conectores de periféricos (EAR, salida de T.V., Conector, etc.). Aquí deducimos que es necesario sumar otro componente, algo que ordene las diferentes vías de comunicación; es decir, que mande la respuesta correcta al, o desde, el periférico adecuado. Porque de lo contrario, el cerebro se enfrentaría a un embotellamiento de información tan grande, que no lo podría entender, y mucho menos procesar.

El que cumple estas funciones de policía de tránsito, es el U.L.A. que, dicho sea de paso, lo fabrica la empresa Ferranti, exclusivamente para Sinclair. Las funciones del U.L.A. son muchas otras más, pero para entender porqué está dentro del ZX-81, alcanza con lo dicho.

Hasta aquí, tenemos definido cómo entra y sale la información en el "cerebro" en forma ordenada, pero, debemos agregar aquí otra cuestión más, y es que esta información debe estar escrita de forma tal que el "cerebro" la pueda entender. Por supuesto, ningún cerebro de ninguna computadora, entienden directamente ni BASIC, ni PASCAL, ni PL1, ni COBOL, ni ningún otro lenguaje de los llamados de "alto nivel". En este punto, podríamos preguntar:

¿Entonces, cómo es que funciona si programamos en BASIC?

La respuesta es muy simple, también existe un intérprete, es decir, un diccionario que traduce, por ejemplo, del BASIC al único lenguaje que cualquier máquina entiende, el "famoso" lenguaje de máquina, código de máquina o Assembler, tales los nombres que recibe. Este diccionario en el ZX-81 es la memoria ROM o memoria sólo de lectura (ROM=Read Only Memory). Se llama así porque no se pueden almacenar datos ni ninguna otra cosa en ella, su contenido es fijo e inalterable. A diferencia de la otra memoria que tiene el computador, el contenido de la ROM no se borra al desconectar la máquina. Es aquí en donde se encuentra definido todo el grupo de senten-

En este artículo veremos lo que a algunos usuarios del ZX-81 les resulta "misterioso", es decir, el manejo de PEEK, POKE, código de maquina, etc., buscando esclarecer estos temas que al primer vistazo nos hacen, muchas veces desconfiar.

cias BASIC de que disponemos para programar, y su correspondiente traducción al código máquina y viceversa.
Más arriba nos referiremos a "la otra memoria". Además de la ROM, existe también la memoria RAM o de acceso aleatorio (RAM=Random Access Memory), y ésta es la memoria en la que se almacenan nuestros programas, o sea es la memoria disponible para el usuario.
Sabiendo todo ésto, podemos decir entonces que, si tenemos 8 K de memoria ROM y 2 K de memoria RAM, entonces nuestra máquina posee 10 K de memoria total, y es completamente cierto.
Con todos estos conocimientos, podemos ahora adentrarnos, de a [illegible]
Todos imaginamos que dentro de la computadora, todos los datos están formados por pulsos eléctricos. Para mayor sencillez, decimos que la presencia de un pulso eléctrico puede representarse por un 1 y la ausencia de pulso pueda representarse por un 0. Esto significa que todos los datos en la computadora pueden representarse por números EXCLUSIVAMENTE formados por unos y ceros. Los números que sólo utilizan ceros y unos se llaman números BINARIOS.
Antes de analizar estos números, es más práctico repasar algunas ideas sobre los números decimales que usamos todos los días. Por ejemplo el número 947 utiliza los tres dígitos 9, 4 y 7, pero como están colocados en un orden establecido, sabemos que el 9 son centenas, el 4 decenas y el 7 unidades. Podríamos escribir:
[illegible]
o también
$947 = 9 \times (10 \times 10) + 4 \times (10) + 7 \times (1)$
Esto quiere decir que cada posición a la izquierda, es 10 veces más que la anterior. Los números decimales utilizan el factor 10 para cada posición. Dicho de otra manera, los números decimales son en base 10.
Aunque se utilice el 10 como base, se puede emplear cualquier otro número como base. Pero la utilizada siempre en Informática es la base 2. Para el manejo más cómodo y rápido de estos números, se expresan en forma hexadecimal.
Aplicando lo visto, podemos "desarmar" un número binario de la siguiente forma.
$0101 = 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$
o también
$0101 = 0 \times (2 \times 2 \times 2) + 1 \times (2 \times 2) + 0 \times 2 + 1 \times 1$
[illegible] un dígito binario es dos veces más que la anterior.
Un número hexadecimal lo podemos desglosar así.

DESENTRAÑANDO LOS CODIGOS

831 H = 8 × 256 + 3 × 16 + 1 × 1
o también
831 H = 8 × (16 × 16) + 3 × (16) + 1 × 1
Aquí, cada posición a la izquierda es 16 veces mayor que la anterior. La letra "H" al final de un número indica que ese número es hexadecimal.
Cualquier número decimal puede escribirse con los dígitos 0 al 9; también podemos escribir cualquier número en binario, utilizando los dígitos 0 y 1, y cualquier número en hexadecimal, utilizando dígitos desde 0 hasta un valor de 15. Así podemos realizar la siguiente tabla:

Decimal	Binario	Hexa
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

En los números hexa, del 10 al 15, se utilizan letras, porque de lo contrario sería muy confuso escribir números entre 0 y 15 con dígitos del 0 al 9.

BITS Y BYTES:

El Z-80 para comunicarse utiliza palabras, pero palabras expresadas en números binarios que a diferencia de nuestro idioma, al lenguaje binario usa sólo palabras de igual longitud generadas a partir de las distintas combinaciones de ceros y unos. El ZX-81 por razones constructivas usa palabras binarias de ocho dígitos que van desde el 00000000 hasta el 11111111, 0 y 255 en decimal, 0 y FF en hexadecimal.
Cada una de las palabras de ocho dígitos se llama BYTE y cada uno de esos ocho dígitos se llama BIT. Es decir que un BYTE contiene, o está formado, por ocho BITS. Es esta la razón por la que el ZX-81 es un computador de 8 bits, porque la extensión de sus palabras tiene 8 dígitos (bits).
El valor de un byte puede utilizarse para representar números, caracteres, órdenes, comandos, etc. Todo depende en que lugar de la memoria se encuentren almacenados.

MEMORIA, PEEK Y POKE:

La memoria del Sinclair la podemos imaginar como una gran cajonera en la cual dentro de cada cajón puede entrar un solo número. Cada uno de estos cajones se llaman posiciones o direcciones de memoria y en ellos puede ser almacenados un solo byte (por supuesto de 8 bits). Las posiciones de memoria van numeradas correlativamente desde cero.
Por otra parte, cuando hablamos de "K" bytes nos referimos a la cantidad de cajones que tiene una determinada memoria, multiplicada por la constante K = 1000. Estrictamente hablando para los computadores K = 1024. Por lo dicho más arriba, si nos encontramos con una memoria de 16 K bytes sabremos que esa memoria puede almacenar:
16 x 1024 = 16384 palabras binarias.
El valor numérico decimal que representa un determinado número binario, almacenado en una cierta posición o dirección de memoria es lo que nos muestra el ZX-81 cuando ejecutamos la instrucción PEEK. Así PEEKear una dirección es "ver" cuál es el código que está almacenado en ese lugar de la memoria. La instrucción es la siguiente:
PEEK, dirección
p.ej: PRINT PEEK, 16418
Devolverá un 2 (en algún caso dará un valor mayor). Esto significa que en la dirección de memoria 16418 está contenido el código 2, en binario por supuesto; el Sinclair lo traduce a decimal para mostrarlo en pantalla.
POKE es el opuesto a PEEK, la instrucción POKE m,n nos permite depositar en la dirección de memoria "m" el Código "n".
Ahora bien, todas las instrucciones de programa son traducidas a binario por la ROM y almacenadas en las direcciones libres de la RAM para ser ejecutadas.
Significa que cuando se ejecuta un programa lo que realmente hace el Z-80 es procesar la información que se encuentra en esas posiciones de memoria en forma de números binarios, y los resultados de ese procesamiento los da también en binario, por lo que la ROM debe volver a actuar aquí para "traducir" ahora del binario al BASIC.
Este es el motivo de la lentitud de los programas escritos en BASIC. En cambio los programas en código máquina, son escritos directamente en código binario (o hexadecimal), de manera que se almacenan directamente y los resultados "salen solos" sin que intervenga traductor alguno. He aquí el porqué de la velocidad de operación de los programas escritos en código máquina.
Existen también dentro de la ROM programas enteros escritos en código que le dicen al Z-80 qué debe hacer. Hay también zonas de variables, variables del sistema, pero nos referiremos a todo esto en otros artículos.
Volvamos a nuestro ejemplo. En la dirección 16418, se encuentra almacenado un cierto valor (usualmente 2) que indica el número de líneas que se reservan en la parte inferior de la pantalla para los mensajes del sistema.
Podemos ver cual es el valor actual haciendo:
PRINT PEEK, 16418
Si ahora colocamos el código 0 en esa dirección haciendo.
POKE 16418,0
Tendremos disponibles para PRINT esas dos últimas líneas.
Se puede probar así.

```
10 POKE 16418,0
20 FOR A=0 TO 23
30 PRINT AT A,10;"LINEA=";A
40 NEXT A
```

POKEando un valor mayor que 2, limitamos el número de líneas que se mueven en un SCROLL:

```
50 POKE 16418,10
60 FOR A=1 TO 4
70 SCROLL
80 NEXT A
```

Para terminar probamos.
PRINT PEEK 16396 + 256 * PEEK 16397
Esto nos dará cuenta RAM tenemos ocupada con un programa y
PRINT PEEK 16404 + 256 * PEEK 16405
nos da la primera dirección de memoria libre
Prueben en sus máquinas PEEKear cualquier dirección para desentrañar los códigos secretos de la Sinclair.
En próximas notas volveremos con estos temas.

C U R S O S

MUSICO

COMP: CZ-2000/TK-90X
CONF: 48 K
CLAS: ENT

Con este programa no sólo podremos generar música, sino también almacenarla en memoria, repetir cuantas veces se nos ocurra la melodía almacenada y grabarla en cinta.

Para cargarlo primero hay que teclear el listado 1, grabarlo en cinta con la instrucción SAVE "MUSICO"

```
5135 PRINT AT 20 1
5140 PRINT AT 21 1 ...
5145 FOR ...
5150 PLOT ...
5155 NEXT
5160 PRINT ...
5165 PAUSE 0
5170 RETURN
5180 REM
5200 PAPER 5 INK 0
5220 GO SUB 2000
5230 PRINT ...
5240 PRINT ...
5250 PRINT ...
5260 PRINT ...
5270 PRINT ...
5280 PRINT ...
5290 PRINT ...
5300 PRINT ...
5310 PRINT ...
5320 PRINT ...
5330 RETURN
6010 REM
6020 GO SUB 4500
6030 PRINT AT 1 17 FLASH 1 ...
6040 PAPER 5 INK 0
6050 LET ...
6060 LET ...
6070 GO SUB 6500 ...
6075 GO SUB 6500
6080 PRINT AT 1 17 FLASH 1 ...
6090 RETURN
6510 LET ...
6520 IF PEEK ... THEN BEEP ...
6530 IF PEEK ... THEN PAUSE ...
6540 IF INKEY$= ... THEN GO TO 650 0
6550 LET q$=INKEY$
6560 IF q$= ... THEN RETURN
6570 IF q$= ... THEN LET ...
6580 IF q$= ... THEN LET ...
6590 IF q$= ... THEN LET ...
6600 LET a=a+2: IF a=... THEN RETURN
6610 LET a=NOT a: IF a THEN LET ...
6620 GO TO 6520
6630 REM ...
7010 PAPER 1: INK 7: GO SUB 8000
7020 PRINT " A save memoria musical completa"
7030 PRINT " B save parte de la memoria musical"
7040 POKE 23658,8
7050 IF INKEY$ ... AND INKEY$ ... THEN GO TO 7050
7060 IF INKEY$= ... THEN LET ...
7070 PRINT ...
7080 INPUT "Menor valor del contador" ...
7090 LET ...
7100 PRINT ...
7110 INPUT "Mayor valor del contador" ...
7120 LET ...
7130 CLS
7140 LET ...
7150 PRINT TAB 8 "Arranque el grabador"
7160 PRINT ...
7170 PRINT AT 1 17 FLASH 1 ...
7180 SAVE ... CODE ...
7190 GO SUB 8200
7200 PRINT AT 1 17 FLASH 1 ...
7210 RETURN
7500 REM ...
7510 PAPER 6: INK 7: GO SUB 8000
7520 PRINT AT 1 17 ...
7530 PRINT AT 12 4 "Pulsa PLAY en el cassette"
7540 LOAD "" CODE
7550 GO SUB 8200
7560 PRINT AT 1 17 FLASH 1 ...
7570 RETURN
7580 REM ...
8000 FOR ...: PRINT AT 1 0 ...
8010 NEXT ...
8020 PRINT AT 2 0
8030 RETURN
8040 REM ...
9999 STOP
```

LINE 70. Luego teclear en la máquina RANDOMIZE USR 0, para provocar un NEW total.

A continuación teclear el listado 2, hacer RUN. Una vez que este programa corra y aparezca el cartel de: STOP STATEMENT, grabarlo a continuación del anterior, utilizando para ello la línea 70.

Este segundo listado es el programa cargador de código de máquina, con su correspondiente listado de códigos puestos en sentencias DATA.

Grabados los dos listados primero el MUSICO, a continuación el listado 2 (beepcode), rebobinamos la cinta, tecleamos RANDOMIZE USR 0, hacemos un LOAD "" normal y los dos programas se cargarán normalmente.

Respecto de las instrucciones, son claras y aparecen en pantalla. Sólo debemos responder a las preguntas u opciones que nos presenta el computador.

```
10 ...
20 ...
30 ...
40 ...
50 ...
60 ...
70 ...
80 ...
90 ...
100 DATA ...
101 DATA ...
102 DATA ...
103 DATA ...
104 DATA ...
105 DATA ...
106 DATA ...
107 DATA ...
108 DATA ...
109 DATA ...
110 DATA ...
111 DATA ...
112 DATA ...
113 DATA ...
114 DATA ...
115 DATA ...
116 DATA ...
117 DATA ...
118 DATA ...
119 DATA ...
120 DATA ...
121 DATA ...
122 DATA ...
123 DATA ...
124 DATA ...
125 DATA ...
126 DATA ...
127 DATA ...
128 DATA ...
129 DATA ...
130 DATA ...
```

ROTULO



COMP: CZ-2000/TK-90X
CONF: 16 K
CLAS: Utilitario.



Este programa permite crear rótulos de caracteres en pantalla y pasarlos a impresora. Las instrucciones aparecen en pantalla y es muy simple de trabajar.

```
10 BORDER 4: PAPER 4: INK 2: CLS
20 PRINT AT 1 13 ...
30 PRINT ...
35 PRINT AT 11,10 "R O T U L O"
40 INK 2
60 PRINT INK 0 AT 2 5 ...
70 PAUSE 0: CLS
80 INPUT "Altura en pantalla" ...
90 INPUT ...
100 IF LEN a$ ... THEN BEEP 1,-20: GO TO 90
110 LET ...
120 FOR ...
125 IF POINT ... THEN GO TO 150
130 PLOT ...: DRAW ...
140 DRAW ...
150 DRAW ...
160 NEXT ...: NEXT ...
180 INPUT "Escribe algo mas" ...
183 IF ... THEN GO TO 200
185 INPUT "Paso a impresora" ...
190 IF ... THEN GO TO 200
195 COPY
200 INPUT "Limpio la pantalla" ...
210 IF ... THEN CLS
220 GO TO 80
230 CLS
240 PRINT AT 11 17 "Chau" ...
250 ... CLS
260 STOP
```

Pantalla

K-64 ROTULO



TARANTULA

COMP: CZ-2000/TK-90X/TS-2068
CONF: 16 K
CLAS. ENT



Es un juego de destreza y cálculo simple. Hay que transportar la mayor cantidad de tesoros en el menor tiempo, sin que nos atrapen las tarántulas.
Este programa utiliza los gráficos definidos por usuario (U.G.D.).
Al pie del listado está la aclaración del caracter que corresponde al gráfico contenido en el listado.

Gráfico.

Pantalla

```
   7 REM
  10 RESTORE
  20 FOR I=1-4 TO 154: FOR x=0 TO 7: READ F: POKE USR CHR$ I+x,F: NEXT X: NEXT I
  30 DATA 24,24,60,90,24,36,36,60
  40 DATA 16,56,84,124,170,170,170,40
  50 DATA [illegible]
  60 DATA [illegible]
  70 DATA [illegible]
  80 DATA [illegible]
  90 DATA [illegible]
 100 DATA [illegible]
 110 DATA [illegible] 16,16,16
 120 DATA 145,82,52,24,16,16,16,16,16
 130 DATA 144,80,48,16,16,16,16,16
 950 CLS
1000 BORDER 4: INK 2: PAPER [illegible]
1020 INPUT "Instrucciones? ";I$
1030 IF I$="S" OR I$="s" THEN GO SUB 9000
1050 LET CNV=134: LET BB=0: LET S1=1: LET SCC=0: LET U=0: LET A=18: LET B=1: LET S2=16: LET S4=1
     LET S2=16: LET S3=16: LET V=9
     LET SC=0: LET TM=0: LET TC=1
     LET R1=1: LET R4=-1: LET R2=-1
     LET R3=1: LET LI=3
1060 [illegible]
1070 PRINT AT 0,2; INK 2;" TARANTULA "
1080 PRINT AT 2,4; INK 2;"VIDAS-";LI;" MARCADOR- ";SC
1090 PRINT AT 1-2,0; INK 1;[illegible]
1100 PRINT AT 12,0;[illegible]
1110 PRINT AT A,B; INK 0;[illegible]
1120 FOR X=18 TO 14 STEP -1: PRINT AT X,0; INK 2;[illegible]: NEXT X
1130 FOR X=5 TO S1-2: PRINT AT X,6; INK 0;"|": NEXT X: PRINT AT S1,6; INK 2;[illegible]
1140 FOR X=5 TO S2-1: PRINT AT X,13; INK 2;[illegible]: NEXT X: PRINT AT S2,13; INK 2;[illegible]
1150 FOR X=5 TO S3-1: PRINT AT X,18; INK 2;[illegible]: NEXT X: PRINT AT S3,18; INK 2;[illegible]
1160 FOR X=5 TO S4-1: PRINT AT X,23; INK 0;"|": NEXT X: PRINT AT S4,23; INK 2;[illegible]
1170 FOR X=1 TO 26: BEEP .05,X: NEXT X
1171 DEF FN O()=INT ((PEEK 23672+256*PEEK 23673+65536*PEEK 23674)/50)
1172 DEF FN [illegible]
1173 DEF FN [illegible]
1180 FOR n=23674 TO 23672 STEP -1: POKE n,0: NEXT n
1990 REM MOVEMENT LOOP
2000 PRINT AT A,B; INK 0;[illegible]
2010 LET BB=BB+(INKEY$="5")-(INKEY$="5")
2015 IF BB=-3 THEN LET BB=-2
2016 IF BB=3 THEN LET BB=2
2017 LET SCC=SCC-2: IF SCC>10 THEN LET SCC=0: LET SC=SC+1
2022 PRINT AT 19,2; CHR$ CNV
2024 PRINT AT 19,31; CHR$ CNV
2026 IF CNV=134 THEN GO TO 2028
2027 LET CNV=134: GO TO 2030
2028 LET CNV=137
2030 PRINT AT 2,4; INK 3;"VIDAS-";LI;" MARCADOR- ";SC
2040 PRINT AT S1,6; INK 0;"|": IF S1=U OR S1/U THEN LET P1=1
2041 IF S1=16 THEN LET R1=-1
2042 IF R1=-1 THEN PRINT AT S1,6;[illegible]
2043 LET S1=S1+R1: PRINT AT S1,6; INK 2;[illegible]
2050 PRINT AT S2,13; INK 2;[illegible]: IF S2=U OR S2 U THEN LET R2=1
2051 IF S2=16 THEN LET R2=-1
2052 IF R2=-1 THEN PRINT AT S2,13;[illegible]
2053 LET S2=S2+R2: PRINT AT S2,13; INK 2;[illegible]
2060 PRINT AT S3,18; INK 2;[illegible]: IF S3=[illegible] THEN LET R3=1
2061 IF S3=16 THEN LET R3=-1
2062 IF R3=-1 THEN PRINT AT S3,18;[illegible]
2063 LET S3=S3+R3: PRINT AT S3,18; INK 2;[illegible]
2070 PRINT AT S4,23; INK 2;[illegible]: IF S4=U OR S4 U THEN LET R4=1
2071 IF S4=16 THEN LET R4=-1
2072 IF R4=-1 THEN PRINT AT S4,23;[illegible]
2073 LET S4=S4+R4: PRINT AT S4,23; INK 2;[illegible]
2080 [illegible]
2081 IF J=50 THEN GO TO 2084
2082 [illegible] 2090
2084 [illegible] -1,3: INK 2;[illegible]
2085 LET J=0: LET [illegible]
2090 IF [illegible] AND [illegible]=13 THEN GO TO 6200
2100 IF [illegible]=13 AND S2=16 THEN GO TO 6100
2110 IF b=18 AND S3=16 THEN GO TO 6200
2115 IF V=19 THEN GO TO 8000
2120 IF b=23 AND S4=16 THEN GO TO 6300
2131 PRINT AT A,B;[illegible]
2132 LET [illegible]
2133 IF B<1 THEN LET B=1
2134 IF B>30 THEN LET B=30
2135 IF B=30 AND TC=1 THEN GO TO 2240
2220 IF B=1 THEN LET TC=1
2230 GO TO 2000
2240 LET TC=0: LET TM=TM+1: LET SC=SC+10: PRINT AT 19-TM,31; INK 2;[illegible]: AT 13+TM,0;[illegible]: IF TM=5 THEN GO TO 6400
2250 GO TO 2000
6000 PRINT AT A,B; OVER 1; INK 4;[illegible]
6010 LET LI=LI-1: IF LI=0 THEN GO TO 8000
6020 LET TC=1
6030 FOR X=1 TO 15: BEEP .005,X: NEXT X
6040 LET A=18: LET B=1: PRINT AT S1,6; INK 2;[illegible]
6050 FOR X=1 TO 15: NEXT X: PRINT AT A,B; INK 0;[illegible]
6060 FOR X=1 TO 15: NEXT X
6070 GO TO 2000
6100 PRINT AT A,B; OVER 1; INK 4;[illegible]
6110 LET LI=LI-1: IF LI=0 THEN GO TO 8000
6120 IF TC=1 THEN LET TC=0
6130 FOR X=1 TO 15: BEEP .005,X: NEXT X
6140 LET A=18: LET B=1: PRINT AT S2,13; INK 2;[illegible]
6150 FOR X=1 TO 15: NEXT X: PRINT AT A,B; INK 0;[illegible]
6160 FOR X=1 TO 15: NEXT X
6170 GO TO 2000
6200 PRINT AT A,B; OVER 1; INK 4;[illegible]
6210 LET LI=LI-1: IF LI=0 THEN GO TO 8000
6220 IF TC=1 THEN LET TC=0
6230 FOR X=1 TO 15: BEEP .005,X: NEXT X
6240 LET A=18: LET B=1: PRINT AT S3,18; INK 2;[illegible]
6250 FOR X=1 TO 15: NEXT X: PRINT AT A,B; INK 0;[illegible]
6260 FOR X=1 TO 15: NEXT X
6270 GO TO 2000
6300 PRINT AT A,B; OVER 1; INK 4;[illegible]
6310 LET LI=LI-1: IF LI=0 THEN GO TO 8000
6320 IF TC=1 THEN LET TC=0
6330 FOR X=1 TO 15: BEEP .005,X: NEXT X
6340 LET A=18: LET B=1: PRINT AT S4,23; INK 2;[illegible]
6350 FOR X=1 TO 15: NEXT X: PRINT AT A,B; INK 0;[illegible]
6360 FOR X=1 TO 15: NEXT X
6370 GO TO 2000
6400 LET TM=0: LET TC=0
6420 FOR X=18 TO [illegible] STEP -1: PRINT AT X,0; INK 2;[illegible]: AT X,31;[illegible]: BEEP .005,X: NEXT X
6430 GO TO 2000
7990 REM SCORE BOARD
8000 FOR X=1 TO 5: NEXT X
8001 FOR X=1 TO 15: BEEP .005,X: NEXT X
8010 CLS
8020 PRINT AT 10,2;"MARCADOR-";SC;" EN ";FN O();" SEGUNDOS                      Quieres intentarlo de nuevo?"
8025 FOR X=-15 TO 40 STEP 5: BEEP .05,X: NEXT X
8030 INPUT I$
8040 IF I$="S" OR I$="s" THEN GO SUB 1000
8070 FOR X=40 TO -40 STEP -5: BEEP .05,X: NEXT X
8080 CLS
8150 STOP
9000 CLS
9010 PRINT AT 0,3;"  TARANTULA  "
9030 PRINT AT 5,0;"  TU OBJETIVO ES TRANSPORTAR EL MAYOR NUMERO DE TESOROS DE UN LADO A OTRO."
9040 PRINT: PRINT: PRINT "EVITANDO LAS TARANTULAS SE ACERCAN."
9050 PRINT: PRINT: PRINT
9060 PRINT "USA LAS TECLAS 5 Y 8 PARA MOVERTE."
9070 PRINT: PRINT
9080 PRINT #1;"Para empezar pulsa cualquier tecla"
9090 PAUSE 0
9100 RETURN
9999 SAVE "TARANTULA" LINE 1
```

II CONCURSO TRIMESTRAL

Para Usuarios de Microcomputadores



PATROCINA



1er. Premio: Una Consola COMMODORE 64
2do. Premio: Una Consola SPECTRUM
3er. Premio: Dos Pasajes a Bariloche
4to. Premio: Un Grabador para Computadora
5to. Premio: Una Impresora Alpha Com 32

Condiciones para participar en el certamen:

1: Los programas deberán ser originales e inéditos, pudiendo cubrir todas las áreas: educativos y de cálculo, uso comercial, entretenimiento, personal y utilitarios para programación, hasta 64 K. 2: El criterio de elección se basará en: originalidad de la idea, método de programación, efectos gráficos y/o sonoros, documentación, presentación y ahorro de memoria. 3: Se enviarán a K-64 grabados en un cassette y acompañados por el listado correspondiente con pantallas y explicación sobre la utilidad y manejo del programa. 4: Puede remitirse más de un programa por cassette, en lo posible grabados dos veces, para mayor seguridad. 5: El concurso K-64 se realizará trimestralmente. El cierre de la recepción de los trabajos será el 15/8/85. 6: K-64 se reserva el derecho de publicación de los programas enviados (como asimismo de la devolución del material recibido).
El cassette deberá ser enviado con su caja y con los datos del programa y del autor, como así también de la computadora para la cual está destinado.

SELECCION MENSUAL

Mensualmente se seleccionarán 60 Programas, los que se harán acreedores a los siguientes premios: Calculadoras, Máquinas de fotos, cassettes con programas, cassettes vírgenes, Becas para Cursos, etc.

Los Programas seleccionados continúan en Concurso para la gran final Trimestral. Presentando este cupón obtendrán un 10% de descuento de las compras que realicen en SANWA S.A. y un 60% para los cursos en EPI.

RETIRO DE CUPONES: K-64: Cerrito 1320 · SANWA: Av. Corrientes 2198, Florida 683 · EPI: Suipacha 848 · 1er. piso, Viamonte 1479 · 8º Piso "B", Florida 683, Av. Corrientes 2188 · Radio del Plata · Av. Santa Fe 2043

AUSPICIAN





SUSCRIPTORES

GANADOR CZ1000 PRIMER SORTEO

FEDERICO KUHNKEN

VILLA GESELL - PCIA. BUENOS AIRES

FELICITACIONES

DEBERAS RETIRARLA EN NUESTRAS OFICINAS CON LA PRESENTACION DEL DOC. DE IDEN.

GANADORES

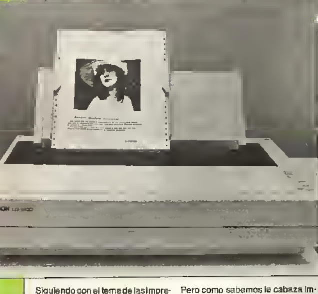
SORTEO - ENCUESTA K64

CASSETTES: MARILENA BARBARO, EDUARDO BRAGA, GERVASIO MARQUES, CINTHIA KAPLANA, JUAN C. SERRA, LUCIANO CHEDRERE, BERNARDO DONY, HECTOR E. VANOTTI, FERNANDO FERRARO, ALEJANDRO MORI, RICARDO GORE, GABRIEL CHERRO, SILVINA NARARI, RAMIRO SELIMAN CLAUDIA BELEMI, NELSON PEREZ, DIEGO FISCHER, FELIX CURANI, ANALIA ALFARO,

BECAS CURSOS BASIC: NERI CIGOI, MARCELO BARDOLI, EZEQUIEL RAMOS, CRISTIAN D'ONOFRIO, ANTONIO ARENAS, JORGE LOPETEGUI, FABIAN DACAL, GABRIEL SONTACHI, GUSTAVO RAVICINI, LUIS VISCIARELLI.

Los premios podrán retirarse en la Administración de K 64 Cerrito 1320, piso 1º, Capital, con documentos de identidad, en el horario 10 a 12 y 15 a 17 horas. Quienes viven en el Interior del país, [illegible] mios por correo

La Impresora



Siguiendo con el tema de las impresoras matriciales, ya señalamos que una característica importante era su velocidad de impresión.

Esta velocidad se mide en C.P.S., que significa **Caracteres Por Segundo**, y una impresora matricial puede tener normalmente 180 a 220 C.P.S.

Pero ojo, esta velocidad debe ser medida imprimiendo Caracteres Normales, si no pierde validez, ya que si la medimos imprimiendo caracteres reducidos la velocidad parecerá más rápida, pero no se puede comparar con otras.

Otra advertencia que hay que hacer con esta velocidad es que se refiere a una impresión continua. Pero como sabemos la cabeza impresora debe volver al escribir cada renglón. Sin embargo, existen impresoras que escriben también al volver. Estas impresoras se llaman "bidireccionales", y por lo tanto aún teniendo los mismos C.P.S. que una "monodireccional", son más rápidas.

En estos casos adquiere mayor importancia otra velocidad, medida en L.P.M., Líneas Por Minuto, que reúne las dos características. Esta característica que no siempre está consignada en los folletos, es importante para elegir una impresora que haga trabajo pesado.

Pero veamos porqué decimos que estas impresoras son mucho más rápidas que las de margarita.

Impresora MARGARITA ó DAISY WHEEL

El principio de funcionamiento de este impresora es totalmente distinto al anterior. Aquí no existen agujas impresoras y un carácter almacenado en ROM. Cada carácter está dibujado en los pétalos de una margarita.

Veamos, margarita se le dice a una rueda que tiene muchos rayos radiales y en el extremo de estos rayos (pétalos) escritos una letra. El modo de funcionamiento consiste en que al reconocer al carácter que queremos imprimir, la "lógica" de la impresora, gira hacia la derecha ó hacia la izquierda la rueda, hasta encontrar el "pétalo" que contiene la letra, y en ese preciso momento dispara un martillo que golpea la letra y la imprime sobre el papel.

Esta búsqueda "lógica" se realiza teniendo en cuenta la posición anterior de la rueda y a lo sumo puede significar un giro de 180 grados de la margarita.

Pero a pesar de ésto es un trabajo lento que demora la impresión. Por lo tanto es normal que la velocidad de esta impresora sea de 18 a 25 C.P.S., 10 veces menor que las matriciales. Sin embargo tienen importantes ventajas:

- La letra no está formada por puntos, es maciza.
- Se puede cambiar la margarita y cambia la letra completamente.

Estas dos características las hacen ideales para escribir cartas ó documentos, donde es más importante la calidad de la impresión que la rapidez.

Existen otras características de las impresoras que son importantes independientemente del tipo. Por ejemplo: el ancho del carro de impresión, que generalmente está dividido en dos tamaños, el de 15" que permite imprimir 132 caracteres en tamaño normal y el de 9" que permite imprimir sólo 80 caracteres.

Otra característica importante es si permite usar hoja común o si solamente se puede usar el formulario contínuo. Parece tonto, pero los costos son diferentes.

Bueno, con esto hemos tratado de dar una rápida y breve descripción de las principales impresoras para microcomputadoras.

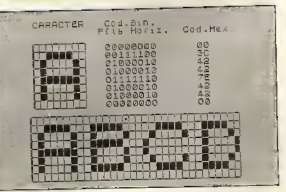
INGENIERO NINO MORENO

LA SENTENCIA PLOT

GRAFICOS CON ALTA RESOLUCION EN LA CZ-1000

La Visualización:

La generación de imágenes en el microcomputador CZ-1000 se realiza en tres etapas. En la primera se generan las líneas en blanco del margen superior de la pantalla. En la segunda se trazan las líneas correspondientes a la parte en que aparecen las líneas de texto.
En la tercera se generan las líneas blancas del margen inferior. Estas operaciones son realizadas por las rutinas de las ROM encargadas de esta tarea. Cuando el ordenador trabaja en modo SLOW, estas rutinas de visualización entran en acción mediante interrupciones periódicas.
Cada caracter se visualiza en pan-



La sentencia Plot permite realizar gráficos con una resolución de 64 x 44 puntos, cada uno con grosor igual a cuatro líneas de borrido. Este programa permite incrementar la resolución a 192 x 178 puntos.

Listado 1

```
Direc-   Cadigos de
clones   operacion          Mnemonicos        Comentarios

4082     3E1E        START  LD A, 1E
         ED47               LD 1, A           Registro 1 llato para re-
         0EFE               LD C, FE          torne a BASIC, el proce-
         067F               LD B, 7F          de.
         ED78               INC A, (C)        Inicia pulso sinc. cuadro
         CB47               BIT 0, A          Retorno a BASIC si se
         C8                 RET Z             pulsa Break.
         3E08               LD A, 08          Registro 1 apunte a zona
         ED47               LD 1, A           0800-09ff.
         0659               LDB, 59
4095     10FE        LAZO1  DJNZ, LAZO1       Primer ciclo de demora.
         DD2100E1           LD 1X, ARCH       1X apunta la, fila carac-
         110000             LD DE, 0000       teres (con bit 15 a "1").
         218430             LD HL, 3C84       Carga No. de lineas a tra-
         A7                 AND A             zar por encima y debajo
         D8                 RET C             de la Imagen, y No. de li-
         D8                 RET C             neas de imagen.
         00                 NOP               Instrucciones de demora.
         7E                 LD A, (HL)
         0619               LD B, 19          No. de caracteres por fila
         4C                 LD C, H           (incluido el C9 final).
         CDBC40             CALL GEN          Generacion de lineas
         58                 LD E, 8           superiores de la pantalla.
         4D                 Ld C, L
         CDBC40             CALL GEN          Generacion de las lineas
         5A                 LD E, D           de Imagen.
         4C                 LD C, H
         CDBC40             CALL GEN          Generacion de las lineas
         060B               LD B, 0B          inferiores de la pantalla.
40B8     10FE        LAZO2  DJNZ, LAZO2       Segundo ciclo de demora.
         18C6               JR, START
40BC     DBFE        GEN    IN A, (FE)        Inicia sincr. de linea
         D3FD               OUT (FD), A       Finaliza pulso de sincr.
         CDCC40             CALL LIN          Genera linea en curso.
         7E                 LD A, (HL)        Instrucciones de demora.
         7E                 LD A, (HL)
         0D                 DEC C             C cuenta No. de lineas.
         C8                 RET Z             por trazar.
         00                 NOP               Instruccion de demora.
         DD19               ADD IX, DE        1X apunta a la siguiente
         18F0               JR GEN            lineas de caracteres.
40CC     DDE9        LIN    JP (IX)           Salto a siguiente linea
                                              de caracteres.
```

Listado 2

```
Direc-    Codigos de
ciones.   operacion.        Mnemonicos        Comentarios

40D2      210061     BORR   LD HL, ARCH
          0EB4              LD C, B4          No. de lineas de imagen.
40D7      0618       LAZO4  LD B, 18          No. de car. por linea.
40D9      369E       LAZO3  LD (HL), 9E       Llena cada linea con
          23                INC HL            codigos de "blanco".
          10FB              DJNZ, LAZO3
          36C9              LD (HL), C9       Coloca codigo de RET
          23                INC HL            al final de cada linea.
          0D                DEC c
          20F3              JR NZ, LAZO4
40E4      C9                RET
```

talla disponiendo pautas de puntos en una matriz de 8X8, de modo que cada fila de caracteres exige el trazado de 8 líneas (fig. 1).

Las pautas de puntos que generan cada caracter se encuentran en el generador de caracteres al final de la ROM, entre las posiciones 1E00 h y 1FFF h. codificadas según la forma en que se muestra la figura 2. Cada caracter se compone de 8 filas horizontales, cada una de las cuales contiene una pauta de 8 puntos. El número binario obtenido disponiendo ceros y unos en la misma forma en que se disponen los puntos blancos y negros de cada fila, es lo que se encuentra en el generador de caracteres

El texto que aparece en pantalla se almacena en una zona de la RAM Llamada "archivo de pantalla" ("display file"), codificado en la forma en que se describe en el apéndice A del manual. Cada fila de caracteres termina con un código de ENTER (76 h). A continuación el CPU busca la primera dirección en el archivo de pantalla, el ULA analiza esta información y se la suministra en serie al modulador de TV, previo paso por el generador de caracteres. Así es analizado cada byte hasta que se encuentre un ENTER de final de línea. El código de ENTER se interpreta como una instrucción HALT (también de código 76 h), lo que hace que al CPU se detenga esperando el inicio de la siguiente línea de barrido.

Alta Resolución:

El programa de CM del listado 1, funciona de una forma parecida a como lo hacen las rutinas de visualización de la ROM.

Para evitar que estas interrumpan el programa de alta resolución, la ejecución deberá realizarse en modo FAST.

Puede obtenerse una zona de archivo en blanco, haciendo actuar la rutina de borrado del listado 2. Esto crea 180 filas con 24 caracteres blancos (código 9E h) cada una.

LA SENTENCIA PLOT

Cómo introducir el Código Máquina:

El programa en CM debe introducirse en una sentencia REM el principio, en las direcciones que se indican en los listados 1 y 2. Puede usarse el programa cargador del listado 3. La sentencia REM deberá contener un mínimo de 99 caracteres.

Se ejecuta el programa con RUN y se añaden las líneas 1000, 1010 y 1020 que se muestren en el listado 4 para comprobar el funcionamiento del CM. Pulsar RND USR 16594 para crear la zona de archivo haciendo actuar la rutina de borrado. Luego introducir POKE 24832,00 y pulsar RUN 1000. Si todo es correcto, aparecerá una pequeña línea vertical cerca del ángulo superior izquierdo de la pantalla, síntoma de que el programa funcione. Interrumpir la visualización con BREAK.

Uso del programa CM:

El listado 5, es un listado BASIC que hace uso del programa CM para representar funciones matemáticas en alta resolución.

La ejecución del programa comienza con el siguiente menú:

1. INICIAR Y BORRAR
2. INTRODUCIR CURVA
3. REVISUALIZAR
4. CENTRADO HORIZONTAL
5. MODIFICAR NUMERO DE LINEAS

Debe comenzarse siempre por la opción 1., que se encaga de crear una zona de archivo en blanco. Puede usarse esta opción para borrar gráficas ya introducidas.

La opción 2., pregunta primero si debe borrar o no las gráficas introducidas, y luego informa cómo hacer para introducir una función nueva: hay que colocarla en forma explícita en la línea 3000 del programa en la forma:

3000 LET Y = f (x)

El programa contiene inicialmente en esta línea el siguiente ejemplo:

3000 LET Y = 85 + 70 * SIN (X / 30)

que sirve para visualizar una curva senoidal.

Una vez hecho ésto, arrancamos con RUN 1000. Con esto, el aparato se pone en FAST y calcula durante unos segundos, colocando los códigos se pasa automáticamente a la visualización de la función. Para interrumpir ésto, pulsar BREAK y volver al menu con RUN.

La opción 3 se limita a volver a mostrar la función.

La opción 4 permite desplazar la gráfica de izquierda a derecha de la pantalla. El desplazamiento se obtiene entrando un número entre 0 y 100. Inicialmente el centrado se ajusta en 11.

Un número mayor provoca un corrimiento a la izquierda y uno menor a la derecha.

La opción 5. permite cambiar el número de líneas usadas, distinto de 180. Al usar esta opción, debe recordarse que la primera y la última línea son usadas para establecer los márgenes superior e inferior y que no se puede escribir sobre ellas, y que la modificación de estas líneas borrará el contenido de la función almacenada.

La zona útil de pantalla corresponde a valores de X entre 1 y 192 y entre 1 y 178 para Y.

Los valores de X se asignan en el bucle FOR-NEXT. Los valores de Y que se salen fuera del rango, no aparecen en pantalla.

Es importante tener cuidado con el tipo de fórmula que se carga, ya que si se encuentran operaciones ilegales, la operación se detendrá, mostrando un código de error.

La opción 3. permite ver el pedazo de gráfica calculada hasta el momento.

Los tramos en pendiente, sobre todo si son muy empinados, pueden aparecer algo discontinuos. Esto se soluciona "arreglando" el paso (STEP) de la línea 200 a un valor menor que 1, aunque a costa de un tiempo de proceso mayor.

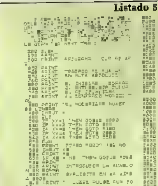
HUGO BUSSO.

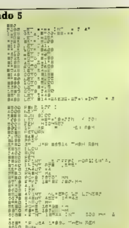
Listado 3

```
  1 REM [illegible]
  5 REM PROGRAMA CARGADOR
 10 LET A$= [illegible]
 20 LET A$=A$+ [illegible]
 30 LET A$=A$+ [illegible]
500 LET D=16514
510 FOR N=1 TO LEN A$ STEP 2
520 LET A=16* CODE A$(N)-28+CODE A$(N+1)-28
530 POKE (D+(N-1)/2),A
540 NEXT N
```

Listado 4

Listado 5





INTERFACE PARA IMPRESORA

ING. PEDRO E. COLLA



Al tope de la lista de prioridades de cualquier usuario de un computador personal que desee sacar el máximo de potencia de una herramienta de este tipo, figura sin ninguna duda el poseer un impresor. Con el mismo se puede acceder a una gama inagotable de aplicaciones útiles en distintos órdenes y, en especial, en lo que se considera la trilogía del software de PC, esto es el procesamiento de la palabra (word processor), la planilla electrónica (electronic spreadsheet) y el almacenamiento de datos para emisión de reportes de distinto tipo.

El impresor es un dispositivo electromecánico que nos permite mantener una salida perdurable para la información sobre una hoja de papel.

Existen impresores de distinto tipo, calidad y precios, y estos factores afectarán en buena medida el uso al cual lo podremos dedicar.

No obstante, se puede separar en principio a los mismos en dos grupos: los que utilizan papel especial y los que utilizan papel común.

En el caso de computador TS2068 el impresor standard para el mismo utiliza papel especial, el cual mediante un tratamiento químico especial permite, mediante la aplicación de calor, imprimir los caracteres que forman el texto. Este tipo de impresión, aparte de no ser totalmente económica, no es apta más allá de trabajos esencialmente caseros, debido a la baja calidad del resultado y al formato normalmente engorroso del tipo de papel utilizado.

Otro tipo de impresor, el que utiliza papel común, es sin duda el que se impone cuando se requiere dar una utilización seria a la salida del computador.

Los impresores de esta última categoría pueden ser de varios tipos pero esencialmente se los puede dividir en dos grupos. Los que imprimen mediante matriz de punto y los que utilizan "margarita" (daisy wheel). Estos últimos de superior calidad de impresión, pero notablemente más costosos.

Debido a ello se han popularizado los impresores de tipo a matriz de puntos (dot matrix), pues representan un compromiso razonable entre la prestación que brindan y su costo.

Existe una gama casi inagotable de variantes dentro de este último tipo. Los hay que imprimen en forma sencilla en un extremo, los que toleran distintas densidades de impresión (caracteres por línea) y tipos de letras en el otro.

Exceptuando la variación en el tipo de prestación, las formas de manejos se mantienen sin embargo parecidas a través de todo el espectro.

Como regla casi general, aquellos que tienen un costo más accesible pueden imprimir hasta 80 caracteres por renglón.

El impresor actúa como periférico del computador, y debido a que no está previsto como tal, será necesario cuando querramos agregarlo utilizar dos elementos: una interfaz de hardware que conecte ambos aparatos electrónicamente y una interfaz de software (driver) que permita al computador el asimilar al nuevo dispositivo en su estructura.

En la presente entrega se describirá la parte correspondiente a la

interfaz de hardware, dejando para una posterior publicación su contrapartida de software.

Los impresores pueden comunicarse con el computador esencialmente en dos formas: en serie y en paralelo.

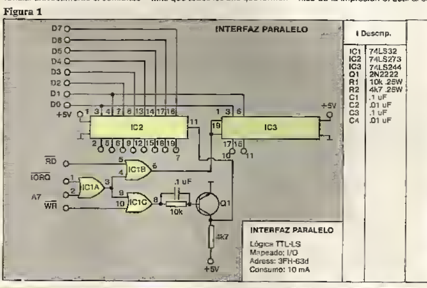
En el primero de los casos, los 8 bits que componen cada caracter por imprimir son enviados uno detrás del otro a través de una única vía. Como la emisión no es continua el canal de comunicación se coloca en estado alto (marca) cuando no hay transmisión. Para diferenciar unívocamente el comienzo y finalización de cada caracter se incluyen bits adicionales al comienzo (start bit) y el final del caracter (stop bit). Los bits son emitidos normalmente a razón de 300 por segundo (300 bauds) lo que representa aproximadamente 30 caracteres en el mismo lapso de tiempo.

A los impresores que utilizan este modo se los conoce con el nombre de RS-232 en alusión a la norma que estipula las condiciones de la comunicación.

Un segundo tipo, en cambio, permite que todos los bits que forman cada caracter (8) se transmitan al mismo tiempo por igual cantidad de canales de comunicación. En este caso, que es el que nos ocupará, la transmisión recibe la denominación de ser en **paralelo** y los impresores son llamados tipo **CENTRONICS** en alusión a la norma que los rige.

Contra lo que se podría suponer, no existen variaciones importantes de velocidad entre ambos tipos, pues este factor en definitiva está limitado por el aspecto mecánico de la impresión el cual al ser

Figura 1



I	Descrip.
IC1	74LS32
IC2	74LS273
IC3	74LS244
Q1	2N2222
R1	10k .25W
R2	4k7 .25W
C1	.1 uF
C2	.01 uF
C3	.1 uF
C4	.01 uF

mucho más lento que el electrónico es el que determina la velocidad final.
El protocolo o forma en que se comunican el impresor y el computador es sencillo y se lleva a cabo mediante como mínimo 10 líneas, 8 de ellas destinadas a los datos y dos a control.
Las líneas de datos reciben la denominación de 00 a 07 y tienen exacta correspondencia con las del bus del computador denominadas D0-D7
Las líneas de control esenciales a la comunicación reciben el nombre de línea BUSY y línea STROBE. La primera de ellas cuando está en un valor alto (+5V) indica que el impresor está ocupado y por lo tanto no está en condiciones de recibir un nuevo caracter porque lo perdería sin poder procesarlo. El estado bajo (0V) de la misma indica como complemento que el impresor está listo (READY) para imprimir un nuevo caracter. Obviamente, esta línea permite que el impresor le "diga" algo al computador.
La otra línea de control (STROBE) tiene dirección contraria y permite que el computador le indique al impresor que lo que en ese momento está presente en las líneas de datos es un caracter válido que debe ser impreso. Esta indicación es aceptada por el impresor cuando esta línea tiene un valor bajo (0V) mientras que cuando asume un valor alto (+5 V), no importa cual sea el contenido eléctrico de las líneas de datos, éstas serán ignoradas por el impresor.
Una interfaz debería permitir que la impresora quede conectada al BUS del computador sólamente cuando tiene lugar una operación de entrada o salida, haciendo que en todo otro momento tal conexión no exista, por lo tanto, no interfiera con el normal desenvolvimiento de las operaciones en el mismo.
Para ello se utiliza la facilidad que permite el microprocesador Z80 de manejar hasta 256 dispositivos de entrada y salida.
Cada dispositivo responderá únicamente cuando su dirección ó número del dispositivo, el cual puede estar comprendido entre 0 y 255, es la que aparece sobre el BUS y no en cualquier otro momento.
El manejo de esta facilidad puede realizarse tanto desde BASIC como desde lenguaje de máquina mediante las instrucciones IN/OUT.
En principio, la dirección que se utiliza puede ser arbitrariamente elegida entre las posibles, pero en este caso en particular se debe tener cuidado de no usar alguna que ya esté utilizando el computador para sus propios fines
En el TS 2068, el teclado, el generador de sonidos y ciertos dispositivos de control utilizen esta facilidad por lo que la elección no es totalmente arbitraria. No obstante, y afortunadamente, todas las direcciones utilizadas tienen la particularidad de ser superiores a la número 128, por lo que utilizando cualquier dirección por debajo de este valor, no existirá peligro de interacción.

Figure 2

PROGRAMA DE PRUEBA DE LA INTERFAZ

```
10 REM
20 REM TEST
30 REM PRUEBA DE LA
   INTERFAZ PARALELO
40 REM IMPRESOR
   CENTRONICS
50 REM
60 FOR I = 1 TO 1000
70 FOR J = 32 TO 127
80 IF IN 63 = 255 THEN
   GOTO 80
90 OUT 63, J
100 NEXT J
110 NEXT I
```

Para nuestro propósito y como una forma de asegurarnos un circuito sencillo, nos aseguraremos unicamente que el impresor responda a cualquier dirección por debajo de la número 128, en todas las operaciones utilizaremos por convención la número 63 (3FH), pero debe quedar claro que cualquier otra causaría el mismo efecto.
Este rango de direcciones cumple con la condición que su bit más significativo (A7) es nulo. Por lo que desde un punto de vista circuital bastará asegurarse esta condición junto con la indicación que se trata de una operación de entrada-salida (línea IORQ en estado bajo) y de que tipo es (lectura o grabación) para tener un interfaz adecuada.
El circuito respectivo es expuesto en la Fig. 1. En la misma se utiliza un integrado 74LS32 (IC1), el cual contiene cuatro compuertas OR; para decidir si se trata de una operación de entrada-salida, si éste tiene la línea A7 baja y cuál es el sentido de dicha operación. La misma, habilita en el caso de que sea una lectura a IC3 el cual es un buffer de tres estados que es utilizado para la lectura de la línea de BUSY previamente descripta.
El integrado IC2 es un octuple Flip-Flop, tipo D, el cual cuando está habilitado toma el contenido de las líneas de datos del bus de la computadora (D0 a D7) y coloca el mismo sobre las correspondientes líneas del impresor. La misma línea sirve para generar la señal de STROBE. Nótese que O1 sirve para invertir la señal unicamente, pues, para la habilitación del impresor requiere un "0" logico (0 V) mientras que IC2 requiere un "1" lógico (+ 5 V).
Debido a la poca cantidad de componentes el circuito puede ser armado sobre una plaqueta experimental mediante técnicas de wire-wrapping o por soldado punto a punto. La alimentación del circuito puede realizarse sin problemas desde la misma computadora dado el bajo consumo del mismo. La conexión debe ser realizada utilizando el conector de expansión que el computador posee en su parte trasera mediante el uso de un conector apropiado. El enlace con la impresora se lleva a cabo mediante un trozo de cable plano de longitud adecuada.
Para unir el cable con la impresora se debe utilizar también un conector apropiado, evitando el soldado directo al impresor, para mantener la posibilidad de desconexión y transporte del mismo.
Debe tenerse especial cuidado en unir todas las masas del impresor (una general y una por cada línea de datos) entre sí y a su vez éstas con la interfaz y el computador si se quiere que el funcionamiento sea correcto, este detalle puede ocasionar que la interfaz no funcione en absoluto.
En la Fig. 2 se expone un pequeño programa en lenguaje BASIC para chequear el funcionamiento del circuito. El mismo imprimirá todo el juego de caracteres del impresor si el conexionado y construcción del circuito es correcto.
En la próxima entrega se brindarán los elementos de software necesarios para integrar completamente computadora, interfaz e impresor.

GLOSARIO DE TERMINOS INFORMATICOS

LETRA "F"

FOREGROUND:
Ejecución automática de programas prioritarios.

FORMAT:
Proceso de codificación inicial a los discos magnéticos antes de que puedan ser utilizados normalmente.

FORTH:
Lenguaje de programación intermedio entre alto nivel y lenguaje de máquina. Utilizado para control y robótica.

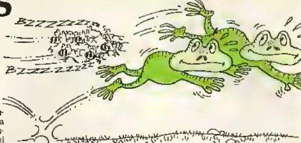
FLOPPY DISK UNIT:
Sistema de almacenamiento masivo de datos que utiliza materiales magnéticos como medio de almacenamiento.

FLOW CHARTS:
Diagramas que ilustran la operación o flujo de un programa, a través del uso de símbolos simples como círculos, rectángulos y rombos, para indicar operaciones. Especialmente útiles para ilustrar relaciones entre varias rutinas de un programa.

RANITAS



Comp: TI 99/4A
Clas: ENT



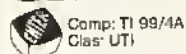
Nos acercamos a un grupo de anfibios y ellos nos piden ayuda para volver a casa. Debemos llegar antes de que las abejas alcancen el borde derecho de la pantalla.

```
100 REM--------------------+
101 REM'                   !
110 REM'  R A N I T A S    !
120 REM'                   !
130 REM'   CLUB  TI/99     !
131 REM'                   !
150 REM--------------------+
160 CALL CLEAR :: PRINT TAB(0);"<<<<<<<>>>>>>>":TAB(0);"< HOME BOUND >":TAB(0);"
<<<<<<<>>>>>>>": : : : : : : : : :
170 CALL SOUND(100,880,0):: PRINT "NECESITAS INSTRUCCIONES?";
180 CALL KEY(0,K,S):: IF S=0 THEN 180 ELSE IF K<>83 THEN CALL CLEAR :: GOTO 250
190 PRINT TAB(10);"HOME BOUND": :"TU TE ACERCASA UN GRUPO DE ANFIBIOS Y E- LLOS
TE PIDEN AYUDA PARA VOLVER A CASA..."
200 PRINT :"CONTROLES: TECLADO": :TAB(12);"PUNTOS": :"SALTAR SOBRE LA TIERRA =
5 PTS":"SALTAR SOBRE LAGO =  10 PTS"
210 PRINT "II NEGROS =  80 PTS":"ABEJAS BLANCAS =  20 PTS": :"LLEGA A CASA ANTES
QUE LAS  ABEJAS ALCANCEN EL BORDE DE-RECHO DE LA PANTALLA"
220 PRINT :"UTILIZA ""S""/""D"" PARA DI-RIGIR ABEJA": :"BUENA SUERTE...PRESIONA
CUALQUIER TECLA";
230 CALL KEY(0,K,S):: IF S=0 THEN 230
240 '
250 DIM CHAR$(1),C$(1,1),M(2),Y(20):: CALL CHARPAT(63,A$):: CALL CHAR(59,A$)
260 CALL INIT :: CALL LOAD(-31878,0):: CALL SPCHIC(N,3,1):: CALL CLEAR :: RANDOM
IZE :: CALL MAGNIFY(3):: N$=RPT$("0",42)
270 CALL CHAR(33,RPT$("F",0),36,"183C6C397E39C3C(1"&N$)
280 CALL CHAR(4C,"7F7EF17F0FF1AF3700000000000000003FF8C5FF9AFFCD7E0")
290 CALL CHAR(44,"02051F181C0F21777F7F3E3F1E1F363620580FC0C10F9C2F7FF7F8EFE9EF836
36",60,N$&N$)
300 CALL CHAR(129,"077800P09000790700000000000000FFFDFFFFFFFFFDFF0")
310 CALL CHAR(132,"7073AEFDF9AE73700000000000000003CE8F8FCFCFBEA3C0")
320 CALL CHAR(136,"3F151F3E3E1F153F00000000000000064AEDEC6CACEAE860")
330 CALL CHAR(140,"30777C7F7F7C773000000000000000070F8BC0C0C8CF8700")
340 C$(0,0)=RPT$("0",29)&"77CE" :: C$(0,1)="00C0E6A63E0F000000000000000000000000
0077CE"
350 C$(1,0)="000000E63E07" :: C$(1,1)="00000E63E070000000000000000000000030794
CF"
360 CALL CHAR(112,"00A25A3C7E9942C3"&N$)
370 CALL CHAR(111,"50A80C0E0E0C6A5"&N$)
380 CALL CHAR(120,"00C342617E3C5A42"&N$)
390 CALL CHAR(124,"0A0630707030D6"&N$)
400 CALL CHAR(96,"105430100083")
410 CALL CHAR(36,"66A5C30000C3A566")
420 CALL CHAR(100,"000000000001F3C346AC380000000000000E0BCF0C148CD858783")
430 RESTORE 1390 :: READ A$ :: CALL CHAR(100,A$)
440 RESTORE 1440 :: READ A$ :: CALL CHAR(92,A$,104,RPT$("F",22)&"07FFFFFFFFFFFFF
F00FF00FFFFFF")
450 FOR I=3 TO 8 :: CALL COLOR(I,2,0):: NEXT I
460 CALL SCREEN(1):: CALL COLOR(1,9,8,3,15,5,9,3,8,10,2,12,11,16,0)
470 '
480 '
490 SC=0 :: SC1=1000 :: S=7 :: F5=4
500 CALL HCHAR(22,1,96,130)
510 CALL HCHAR(1,1,96,64)
520 CALL HCHAR(16,1,104,224):: CALL HCHAR(17,1,108,32):: CALL HCHAR(10,1,106,32)
:: CALL HCHAR(21,1,105,32)
530 CALL LOAD(-31878,0):: FOR I=4 TO 31 STEP 5 :: CALL HCHAR(1,I,33,2):: CALL HC
HAR(2,I,38,2):: NEXT I
540 B=0 :: CALL SPRITE(#27,60,10,1,256)
550 '
560 P=24
570 FOR I=1 TO 11 STEP 5 :: A=24 :: FOR J=I+1 TO I+5 :: CALL SPRITE(#J,40,11,A,1
):: A=A+11 :: NEXT J :: I=I+7 :: NEXT I
580 '
590 A=120
600 FOR I=17 TO 22 :: CALL SPRITE(#I,(I-17)*4+126,INT(RND*13+3),A,1,#I+4,(I-17)*
4+128,INT(RND*13+3),0,INT(RND*200+30)):: A=A+11 :: NEXT I
610 '
620 FOR I=0 TO 1 :: RANDOMIZE :: CALL COLOR(#I+2,11,#I+7,11,#I+12,11):: IF INT(R
ND*2)AND H THEN 660
630 M(I)=INT(RND*4)-1 :: IF M(I)=0 THEN 630
640 B=I/2-INT(I/2):: IF B=0 THEN B=B+1
650 M(I)=M(I)*B*(SC/1000+1)
660 NEXT I :: CALL SCREEN(8)
670 FOR J=0 TO 4 :: CALL MOTION(#J+2,0,M(J),#J+7,0,M(J),#J+12,0,M(J)):: NEXT J
680 '
690 FOR I=17 TO 22
700 RANDOMIZE :: IF INT(RND*3)=0 AND H THEN 710 ELSE B=SGN(I-18.5)*INT(RND*5+4)*
M(I)*(1+SC/2000)):: CALL MOTION(#I,0,0,#I+4,0,0)
710 NEXT I :: GOTO 730
720 FOR I=0 TO 8 :: CALL COLOR(#I+2,11,#I+7,11,#I+12,11):: NEXT I
730 CALL LOAD(-31878,26):: CALL SPRITE(#1,112,16,183,126,#25,36,10,1,16):: IF IN
T(RND*3)>0 THEN CALL SPRITE(#26,100,2,184,1,0,RND*6+3)
740 RANDOMIZE :: B=INT(RND*6):: CALL GCHAR(1,B*5+5,I):: IF I=33 THEN CALL VCHAR(
1,B*5+5,34)ELSE B=-4/5
750 '
760 CALL HCHAR(23,1,96,32):: DISPLAY AT(24,2)SIZE(F5):RPT$(CHR$(112),F5-1)&CHR$(
96)
770 DISPLAY AT(23,12)SIZE(11)BEEP:"LISTO";" :: FOR I=1 TO 1 :: CALL KEY(0,I,ST)::
IF ST THEN I=10
780 NEXT I :: CALL HCHAR(23,1,96,32):: IF I=9 THEN 770 ELSE CALL HCHAR(15,10,11,
15)
790 X=183 :: Y=126 :: CALL MOTION(#25,0,RND*1)
800 '
810 CALL KEY(1,K,S):: CALL POSITION(#25,I1,I1):: IF I1>243 THEN CALL TIMEEND ::
GOTO 1190 ELSE IF ST=0 THEN 920
820 IF K=5 THEN 840 ELSE IF X<104 THEN 960 ELSE X=X-8 :: CALL PATTERN(#1,112)::
CALL LOCATE(#1,X,Y):: GOSUB 910
830 IF HIT THEN 1190 ELSE X=X-8 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-100,400,0):
: GOTO 920
840 IF K=3 THEN 860 ELSE IF Y>240 THEN 900 ELSE Y=Y+4 :: CALL PATTERN(#1,116)::
CALL LOCATE(#1,X,Y):: GOSUB 910
850 IF HIT THEN 1190 ELSE Y=Y+4 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-100,600,0):
: GOTO 920
860 IF K=0 THEN 880 ELSE IF X>180 THEN 880 ELSE X=X+8 :: CALL PATTERN(#1,120)::
CALL LOCATE(#1,X,Y):: IF X>96 THEN GOSUB 910 ELSE HIT=0
870 IF HIT THEN 1190 ELSE X=X+8 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-100,800,0):
: GOTO 920
880 IF K=1 THEN 970 ELSE IF Y<12 THEN 900 ELSE Y=Y-4 :: CALL PATTERN(#1,124):: C
ALL LOCATE(#1,X,Y):: GOSUB 910
890 IF HIT THEN 1190 ELSE Y=Y-4 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-100,530,0):
: GOTO 920
900 CALL SOUND(100,110,5):: GOTO 810
910 CALL COINC(ALL,HIT):: CALL SOUND(-100,220,7):: SC=SC+5 :: RETURN
920 CALL COINC(ALL,HIT):: IF NOT HIT THEN CALL COINC(ALL,HIT):: IF NOT HIT THEN
CALL COINC(ALL,HIT):: IF NOT HIT THEN 810
930 GOTO 1190
940 '
950 CALL KEY(1,I,ST):: CALL POSITION(#25,I1,I1):: IF I1>243 THEN CALL TIMEEND ::
 GOTO 1190 ELSE IF ST=0 OR ST=-1 THEN 950 ELSE CALL SOUND(-100,999,5,-1,2)
960 IF I=5 THEN 970 ELSE CALL MOTION(#1,0,0):: CALL POSITION(#1,X,X):: IF X<24 T
HEN 1020 ELSE CALL PATTERN(#1,112):: CALL LOCATE(#1,X-11,Y):: C=39 :: GOTO 1230
970 IF K=3 THEN 900 ELSE CALL MOTION(#1,0,0):: CALL POSITION(#1,X,Y):: IF Y>240
THEN 1190 ELSE CALL PATTERN(#1,116):: CALL LOCATE(#1,X,Y+8):: C=23 :: GOTO 1230
```


K64

```
990 IF K=0 THEN 990 ELSE CALL MOTION(#1,0,0):: CALL POSITION(#1,I,Y):: IF I<87 T
HEN CALL PATTERN(#1,120):: CALL LOCATE(#1,I+16,I):: D=2 :: GOTO 1250 ELSE 960
990 IF I>2 THEN 1000 ELSE CALL MOTION(#1,A,0):: CALL POSITION(#1,I,Y):: IF Y<12
THEN 1190 ELSE CALL PATTERN(#1,124):: CALL LOCATE(#1,I,Y-8):: C=23 :: GOTO 1230
1000 GOTO 940
1010 '
1020 IF Y>250 THEN 1170 ELSE Y=Y+4 :: CALL MOTION(#25,0,0):: Y=INT(Y/8)+1 :: CAL
L GCHAR(2,I,HIT):: IF HIT=33 THEN 1170 ELSE CALL GCHAR(I,I,HIT):: H=H+1 :: SC=SC
+50
1030 CALL GCHAR(2,I-1,ERR):: IF ERR=22 THEN I=I+1 :: CALL VCHAR(I,Y,HIT)
1040 CALL DELSPRITE(#2):: CALL SOUND(60,-2,0,110,0):: CALL VCHAR(I,Y-1,44):: CAL
L VCHAR(I,Y,46):: CALL VCHAR(2,Y-1,45):: CALL HCHAR(2,Y,41)
1050 IF HIT=34 THEN CALL SOUND(300,880,0,330,3):: CALL SOUND(100,770,0):: CALL S
OUND(300,550,0)ELSE CALL SOUND(1,0*5+5,3)
1060 Y=19-I+8 :: CALL KEY(1,I,I):: IF K=2 AND K=3 THEN 1130 ELSE CALL POSITION(
#25,I,I):: CALL LOCATE(#27,3,Y+16*I+2)
1070 CALL SOUND(-200,-2,3):: CALL CHAR(60,C$(K*2)+1,0):: CALL SOUND(-240,-5,3)
:: CALL CHAR(60,C$(K*2)+1,1)
1080 CALL COINC(ALL,I):: IF I=0 THEN CALL COINC(ALL,I):: IF I=0 THEN 1110
1090 CALL SOUND(50,-5,2,110,3):: SC=SC+200 :: CALL CHAR(36,"0C1E0202040C101E"&RP
T$("0",16)&"4EGAAAAAAAEE44")
1100 FOR I=800 TO 2000 STEP 50 :: CALL SOUND(-99,I,2):: NEXT I :: CALL DELSPRITE
(#25):: CALL CHAR(36,"183C6C3F7E3F6C3C3C"&RPT$("0",15))
1110 IF K=1 THEN CALL CHAR(62,"0")
1120 CALL CHAR(60,RPT$("F",33)):: CALL LOCATE(#27,I,I)
1130 DISPLAY AT(15,10)SIZE(LEN(STR$(SC))+6):"SCORE:"&STR$(SC):: IF HIT=34 THEN D
ISPLAY AT(23,10)SIZE(10):"BONUS:100" :: CALL DING(SC,80)
1140 GOSUB 1270 :: IF S<14 THEN S=S+1
1150 IF H=6 THEN 1180 ELSE CALL SCREEN(6):: DISPLAY AT(23,10)SIZE(10):"BONUS:500
";CHR$(96):: CALL DING(SC,500):: CALL SONG(5):: GOSUB 1270
1160 FOR I=4 TO 31 STEP 5 :: CALL SOUND(200,770-I*10,0):: CALL HCHAR(2,I,33,2)::
 CALL VCHAR(I,I,45):: CALL VCHAR(I,I+1,47)
1170 CALL SOUND(200,770+I*10,0):: CALL HCHAR(I,I,33,2):: NEXT I :: H=0 :: CALL S
ONG(1,A):: GOTO 730
1180 S=14 :: CALL SONG(1):: GOTO 1250
1190 CALL VCHAR(I,B*5+5,33):: CALL MOTION(#25,0,0):: IF I<104 THEN CALL DROWN(FS
)ELSE CALL SQUISH(FS,B,A)
1200 CALL DELSPRITE(#25,#26):: DISPLAY AT(15,10)SIZE(LEN(STR$(SC))+6):"SCORE:"&S
TR$(SC):: GOSUB 1270
1210 IF FS=0 THEN CALL OVER(SC,N$(),0):: GOTO 490 ELSE IF I>14 THEN 620 ELSE C
ALL FROG(FS+2,15):: GOTO 720
1220 '
1230 CALL COINC(ALL,HIT):: IF NOT HIT THEN CALL COINC(ALL,HIT):: IF NOT HIT THEN
 CALL COINC(ALL,HIT):: IF NOT HIT THEN 1190
1240 SC=SC+10
1250 CALL MOTION(#1,0,0):: CALL SOUND(-100,999,7,500,4):: GOTO 750
1260 '
1270 IF SC<BCI THEN RETURN ELSE CALL SCREEN(11):: DISPLAY AT(23,10)SIZE(10):"EXT
RA";CHR$(96);"FROG" :: CALL SOUND(200,440,2):: CALL SOUND(160,550,2):: CALL SOUND(
199,770,0)
1280 CALL SOUND(990,880,2):: CALL SCREEN(4):: SCI=SCI+1000 :: FS=FS+1 :: CALL FR
OG(1,FS+2)
1290 CALL DELSPRITE(#1):: CALL VCHAR(24,FS+3,112):: CALL VCHAR(24,8,96):: RETURN
1300 '
1310 DATA 261,250,349,250,349,250,391,250,391,250,391,250,440,375,440,125,523,25
0,440,250,440,250,440,250
1320 DATA 391,250,391,250,349,500
1330 DATA 261,250,349,250,349,250,391,250,391,250,440,375,466,125,523,250,440,25
0,440,250,440,250
1340 DATA 391,250,391,250,349,500
1350 DATA 523,125,440,125,440,250,440,250,440,250,440,250,440,125,440,125,391,25
0,391,250,391,250
1360 DATA 523,125,466,125,440,250,440,250,440,250,440,250,440,125,440,125,391,25
0,391,250,391,250
1370 DATA 523,250,440,250,349,250,391,250,391,250,349,375,44733,1
1380 'SPLASH
1390 DATA 0000000010202040502010000000000000000000080400020040C
1400 DATA 000104009090A121414221904030000000000E040207040282848C810E
1410 DATA 07001102029440811904000404430090AC02010C034120444902202040080
1420 DATA 106090C4000226100600000409040260002124100040204002000040002142
1430 '
1440 DATA 0000000F1F1F0F07011F0700000000000000000E0E0E0E0E0C0808
1450 DATA 000010392F1F0F040F1F1B4110000000000000000FOF8F8F0E09080C
1460 DATA 0000200097F7F331F1F1F7F3518010000000000E040F8FFF6FEEC0E0A0C0E0C
1470 '
1480 DATA 20,19,182,85,142,145,155,41,251,82,201,56,30,70,78,68,197,155,183,177,
201
1490 '
1500 DATA 22,21,40,82,141,78,78,90 82,28,230,237,112,11,242,168,39,0,1,194,226,2
4,1
1510 'SUBPROGRAMAS
1520 SUB SONG(I):: RESTORE
1530 FOR I=1 TO I :: READ A,Z :: CALL SOUND(0,A,0):: NEXT I :: SUBEND
1540 SUB SQUISH(FS,B,A):: CALL SCREEN(7):: CALL POSITION(#1,A,B):: CALL SPRITE(#2
0,92,2,A-4,B-4)
1550 CALL SCREEN(4):: FOR I=0 TO 30 :: CALL SOUND(-500,150,1,-2,1):: NEXT I :: R
ESTORE 1440 :: FS=FS-1
1560 FOR I=1 TO 3 :: READ A$ :: CALL CHAR(92,A$):: NEXT I :: FOR I=1 TO 300 :: N
EXT I :: CALL DELSPRITE(#1,#20)
1570 RESTORE 1440 :: READ A$ :: CALL CHAR(92,A$):: SUBEND
1580 SUB DROWN(FS):: CALL SOUND(-4000,-6,0):: CALL POSITION(#1,I,Y):: CALL SPRIT
E(#1,108,0,I-4,I):: RESTORE 1390
1590 FOR I=1 TO 4 :: CALL SOUND(-4000,-5,I*5):: READ A$ :: CALL CHAR(108,A$):: N
EXT I
1600 CALL DELSPRITE(#1):: FS=FS-1 :: SUBEND
1610 SUB TIMEEND :: CALL MOTION(#25,0,A)
1620 FOR I=2 TO 8 :: FOR A=2 TO 10 :: CALL COLOR(#A,I):: NEXT A :: NEXT I
1630 CALL MOTION(#1,0,0):: SUBEND
1640 SUB OVER(SC,N$(),N()):: CALL DELSPRITE(ALL)
1650 IF SC<=N(2)THEN 1680 ELSE IF SC<=N(1)THEN I=2 :: N(1)=SC ELSE N(2)=N(1):: N
(1)=N$(1):: IF SC<=N(0)THEN I=1 :: N(1)=SC ELSE I=0 :: N$(1)=N$(0):: N(1)=N(0):
: N(0)=SC
1660 CALL SONG(55):: FOR I=1 TO 8 :: CALL COLOR(I,16,4):: NEXT I
1670 CALL SCREEN(5):: DISPLAY AT(8,10):"ESTE ES EL FIN": :"TU ERES UNO DE LOS M
EJORES": :"TU NOMBRE:" :: ACCEPT AT(12,12):N$(I)
1680 FOR I=3 TO 8 :: CALL COLOR(I,2,8):: NEXT I
1690 DISPLAY AT(2,9):"HIGH SCORE" :: FOR I=0 TO 1 :: DISPLAY AT(I+5,7):N$(I);TAB
(16);N(I):: NEXT I
1700 DISPLAY AT(10,1):(TAB(10);"*** F I N ***": :"PRESI READ O BACK" ) :: DISPL
AY AT(11,10)SIZE(LEN(STR$(SC))+6):"SCORE:"&STR$(SC)
1710 CALL KEY(0,K,ST):: IF K=11 THEN CALL CLEAR :: STOP ELSE IF K<>6 THEN 1710
1720 CALL CLEAR :: CALL SCREEN(11):: FOR K=2 TO 8 :: CALL COLOR(K,2,4):: NEXT K :
: CALL COLOR(1,1,8):: SUBEND
1730 SUB FROG(A,B):: CALL VCHAR(24,A+1,96):: CALL SPRITE(#1,116,14,185,A/8+1)
1740 FOR I=4 TO 2 STEP .5 :: CALL SOUND(-99,690,19):: CALL LOCATE(#1,185,I*B+1)
:: CALL SOUND(-99,220,19):: NEXT I
1750 SUBEND
1760 SUB DING(0,0)
1770 W=A+0 :: FOR I=A TO W STEP 5 :: CALL SOUND(-1,3500,2):: DISPLAY AT(15,16)
SIZE(LEN(STR$(I))):STR$(I):: NEXT I
1780 A=A+3 :: SUBEND
1790 SUB SPCH(A$,S$):: RESTORE 1480 :: S$,A$=CHR$(96)&CHR$(10):: READ A :: FOR I=
1 TO A :: READ B :: S$=S$&CHR$(B):: NEXT I :: READ A :: FOR I=1 TO A :: READ B :
: A$=A$&CHR$(B):: NEXT I :: SUBEND
```

GENERADOR DE CARACTERES



Comp: TI 99/4A
Clas: UTI

Utilicemos este programa para definir un "Sprite" de 4 caracteres, dándole así la salida al programa. Logramos, por otro lado, definición del caracter en código hexadecimal y se lo puede guardar en un diskette formato Merge, como si fuera una línea de data. Obtenemos, también, la posibilidad de imprimir el caracter (abajo del programa). Y, como otras opciones, podemos rotarlo, desplazarlo e invertirlo.

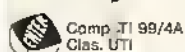


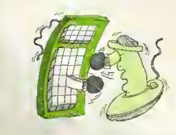

```
100 REM+-----------------+
110 REM'                 '
120 REM'    GENERADOR    '
130 REM'                 '
140 REM'       DE        '
150 REM'                 '
160 REM'   CARACTERES.   '
170 REM'                 '
180 REM'    CLUB TI/99.  '
190 REM+-----------------+
200 DATA 1111111111111111111111111111111111111111111111111111111111111111
210 OPTION BASE 1 :: DIM SP(16,16),B(8),SP1(16,16):: I=20 :: I,C=100 :: LX,LY=8
220 CALL CHAR(100,RPT$("0",16)&RPT$("F",16)&"007E424242427E00FF8181818181FF")
230 HEX$="0123456789ABCDEF" :: CALL COLOR(9,2,15)
240 '
250 CALL MAGNIFY(3):: CALL SCREEN(12):: CALL CLEAR
260 DISPLAY AT(1,3):"AUTO DEFINICION DE SPRITES"::::"  1234567812345678"
270 FOR I=0 TO 1 :: FOR I1=1 TO 8 :: CALL VCHAR(4+I1+8*I,1,48+I1):: NEXT I1 :: N
EXT I
280 FOR I=5 TO 20 :: CALL HCHAR(I,5,1,16):: NEXT I :: C=K :: CALL VCHAR(LX,LY,C+
2)
290 CALL SCREEN(12):: DISPLAY AT(5,22):"PRES" :: DISPLAY AT(7,T):"1 BLANCO" :: D
ISPLAY AT(9,1):"2 NEGRO" :: DISPLAY AT(11,T):"3 OPCION"
300 DISPLAY AT(13,1):"4 PARAR" :: DISPLAY AT(17,T): :: DISPLAY AT(15,1):"5 DATOS
"
310 DISPLAY AT(17,1):"5 CALCULO" :: DISPLAY AT(19,T):"0 ON/OFF"
320 CALL SPRITE(#1,132,8,12,190)
330 CALL KEY(1,I,S):: IF S=0 THEN 330 ELSE I,Y=0 :: IF K=5 THEN I=-1 :: GOTO 670
 ELSE IF K=1 THEN I=1 :: GOTO 670 ELSE IF K=2 THEN Y=-1 :: GOTO 670
340 IF K=3 THEN Y=1 :: GOTO 670 ELSE IF K=0 THEN I,Y=-1 :: GOTO 670 ELSE IF I=4
THEN I,Y=-1 :: GOTO 670 ELSE IF K=14 THEN I,Y=1 :: GOTO 670 ELSE IF K=15 THEN I=
1 :: Y=-1 :: GOTO 670
350 IF K=6 THEN I=-1 :: Y=1 :: GOTO 670 ELSE IF K<>18 THEN 370 ELSE CALL SOUND(1
00,110,0,-4,0):: IF C=101 THEN C=100 ELSE C=101
360 CALL VCHAR(LX,LY,C+2):: GOTO 330
370 CALL SOUND(100,440,0):: IF K=19 THEN K=100 :: GOTO 280 ELSE IF K=16 THEN 710
 ELSE IF K=7 THEN K=101 :: GOTO 280
380 IF K=9 THEN STOP ELSE IF K=8 THEN 400 ELSE IF K<>10 THEN 330 ELSE GOSUB 940
:: GOTO 620
390 '
400 CALL SCREEN(10):: DISPLAY AT(7,T):"1 VMIRROR" :: DISPLAY AT(9,1):"2 HMIRROR"
 :: DISPLAY AT(11,1):"3 REVERSA" :: DISPLAY AT(13,T):"4 ROTAC" :: DISPLAY AT(15,
1):"5 SCROLL" :: DISPLAY AT(17,1):"6 RETORNO"
410 CALL KEY(0,K,S):: IF S=0 OR K<49 OR K>54 THEN 410 ELSE CALL SOUND(100,330,0)
:: H=K-48 :: CALL VCHAR(LX,LY,C):: CALL VCHAR(H*2+5,23,42)
420 IF H=6 THEN 620 ELSE IF H=3 THEN 640 ELSE IF H<>5 THEN 550
430 '
440 CALL SCREEN(6):: DISPLAY AT(7,1):"1 ARRIBA" :: DISPLAY AT(9,1):"2 ABAJO" ::
DISPLAY AT(11,1):"3 IZQ." :: DISPLAY AT(13,1):"4 DER." :: DISPLAY AT(15,1): :: D
ISPLAY AT(17,1):
450 CALL KEY(0,I,S):: IF S=0 OR I<49 OR I>52 THEN 450 ELSE CALL VCHAR((I-49)*2+7
,23,42)
460 '
470 CALL SOUND(100,110,0):: IF I>50 THEN 510 ELSE IF K=50 THEN K=20 :: S=-1 ELSE
 K=5 :: S=1
480 FOR I=K TO S*15+K STEP S :: FOR J=5 TO 20 :: CALL GCHAR(I+S,J,H):: IF H<>100
 AND H<>101 THEN H=100
490 CALL VCHAR(I,J,H):: NEXT J :: NEXT I :: GOTO 620
500 '
510 IF I=51 THEN K=5 :: S=1 ELSE K=20 :: S=-1
520 FOR J=K TO S*15+K STEP S :: FOR I=5 TO 20 :: CALL GCHAR(I,J+S,H):: IF H<>100
 AND H<>101 THEN H=100
530 CALL VCHAR(I,J,H):: NEXT I :: NEXT J :: GOTO 620
540 '
550 FOR I=1 TO 16 :: FOR J=1 TO 16 :: CALL GCHAR(I+4,J+4,SP(I,J)):: NEXT J :: CA
LL VCHAR(I+4,23,42):: NEXT I :: CALL VCHAR(5,23,32,16)
560 'MIRRORS
570 IF H=1 THEN I=4 :: S=-21 ELSE IF H=2 THEN K=-21 :: S=4 ELSE 600
580 FOR I=1 TO 16 :: FOR J=1 TO 16 :: CALL VCHAR(ABS(I+I),ABS(S+J),SP(I,J)):: NE
XT J :: NEXT I :: GOTO 620
590 'ROTACION
600 FOR J=20 TO 5 STEP -1 :: FOR I=5 TO 20 :: CALL HCHAR(I,J,SP(21-J,I-4)):: NEX
T I :: NEXT J
```

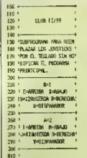



```
610 !
620 CALL GCHAR(LX,LY,C):: CALL VCHAR(LX,LY,C+2):: GOTO 290
630 !REVERSA
640 FOR I=1 TO 18 :: FOR J=1 TO 16 :: CALL GCHAR(I+4,J+4,H):: IF H=100 THEN H=10
1 ELSE H=100
650 CALL VCHAR(I+4,J+4,H):: NEXT J :: NEXT I :: GOTO 620
660 !MOVER CURSOR
670 IF LX+X=21 THEN X=-15 ELSE IF LX+X=4 THEN X=15
680 IF LY+Y=21 THEN Y=-15 ELSE IF LY+Y=4 THEN Y=15
690 CALL VCHAR(LX,LY,C):: LX=LX+X :: LY=LY+Y :: CALL VCHAR(LX,LY,C+8):: GOTO 330
700 !DEFINICION SPRITES
710 CALL VCHAR(LX,LY,C):: CALL SOUND(100,120,0):: CALL SOUND(300,112,0)
720 FOR Q=5 TO 13 STEP 8 :: FOR W=5 TO 13 STEP 8 :: FOR E=0 TO 7 :: FOR R=0 TO 7
 :: CALL GCHAR(Q+E,W+R,CH):: B(E+1,R+1)=CH-100
730 NEXT R :: NEXT E :: GOSUB 860 :: DISPLAY AT(22,1):M$ :: NEXT W :: NEXT Q
740 !
750 CALL CHAR(132,M$):: DISPLAY AT(7,7):"GRABAR" :: FOR I=9 TO 15 STEP 2 :
: DISPLAY AT(I,7): :: NEXT I
760 A$=""
770 CALL KEY(0,KEY,ST):: IF ST=0 THEN 770 ELSE IF KEY<>ASC("S")THEN 830
780 DISPLAY AT(11,1):" NOMBRE?" :: DISPLAY AT(13,7):"LINEA" :: ACCEPT AT(10,1)SI
ZE(10)BEEP:NA$ :: ACCEPT AT(11,25)BEEP VALIDATE(DIGIT)SIZE(4):LI :: IF LI>254 THE
N 780 ELSE IF LI=0 THEN 810
790 OPEN #1:"DSK1."&NA$,VARIABLE 80
800 FOR I=1 TO 64 STEP 16 :: PRINT #1:STR$(LI)&"DATA" ",71,1,71;"DATA ";SEG$(M$
,I,4);",";SEG$(M$,I+4,4);",";SEG$(M$,I+8,4);",";SEG$(M$,I+12,4):: NEXT I :: C
LOSE #1 :: GOTO 830
810 OPEN #1:"DSK1."&NA$,VARIABLE 163 :: PRINT #1:CHR$(0)&CHR$(L)&CHR$(147)&CHR$
(200)&CHR$(LEN(M$))&M$&CHR$(0):: PRINT #1:CHR$(255)&CHR$(255):: CLOSE #1
820 !
830 DISPLAY AT(17,1):"GRABAR" :: DISPLAY AT(19,7):"IMPRESOR?" :: ACCEPT AT(19,14)VA
LIDATE("SN")BEEP SIZE(1):A$ :: IF A$="N" THEN M$="" :: GOTO 620 ELSE IF A$<>"S"
THEN 830
840 GOSUB 1010 :: M$="" :: GOTO 620
850 !
860 FOR R=1 TO 8
870 LOW=B(R,5)*8+B(R,6)*4+B(R,7)*2+B(R,8)+1
880 HIGH=B(R,1)*8+B(R,2)*4+B(R,3)*2+B(R,4)+1
890 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HEX$,LOW,1)
900 NEXT R :: RETURN
910 !
920 DATA 0,0,0,0,0,0,0,1,0,0,1,0,0,0,1,1,0,1,0,0,0,1,0,1,0,1,1,0,0,1,1,1,1,0,0,0
930 DATA 1,0,0,1,1,0,1,0,1,0,1,1,1,1,0,0,1,1,0,1,1,1,1,0,1,1,1,1
940 RESTORE 920 :: FOR O=1 TO 16 :: READ B,E,D,K :: DP$(O)=CHR$(B+100)&CHR$(E+10
0)&CHR$(D+100)&CHR$(K+100):: NEXT O
950 RESTORE 32767 :: READ M$ :: DISPLAY AT(22,1):M$
960 FOR R=3 TO 11 STEP 8 :: FOR O=5 TO 20 :: FOR W=0 TO 7 STEP 8
970 M$=DP$(POS(HEX$,SEG$(M$,1,1),1)):: M$=SEG$(M$,2,LEN(M$)-1):: DISPLAY AT(R,O+
W)SIZE(4):M$
980 NEXT W :: NEXT O :: NEXT R :: RETURN
990 REM
1000 REM
1010 OPEN #1:"RS232.BA=9600,CR"
1020 PRINT #1:CHR$(27);"A";CHR$(8)
1030 FOR R=5 TO 20 :: PRINT #1:CHR$(10);CHR$(27);"K";CHR$(49);CHR$(0);CHR$(27);::
 FOR O=5 TO 20 :: CALL GCHAR(R,O,S):: IF C=101 THEN 1050
1040 PRINT #1:CHR$(0);CHR$(14);CHR$(7):: GOTO 1060
1050 PRINT #1:CHR$(7);CHR$(7);CHR$(7);
1060 NEXT O :: IF R=9 THEN PRINT #1:"  NOMBRE: ";NA$ ELSE IF R=14 THEN PRINT #1:
"  LINEA: ";STR$(LI)
1070 NEXT R :: PRINT #1:CHR$(10);CHR$(27);"K";CHR$(49);CHR$(0);
1080 FOR R=1 TO 49 :: PRINT #1:CHR$(4);:: NEXT R
1090 PRINT #1:CHR$(27);"@";CHR$(10);"HEX: ";M$;CHR$(13)
1100 CLOSE #1 :: RETURN
```

PARA REEMPLAZAR LOS JOYSTICKS







```
100 !+-------------------+
110 !                    !
120 !     CLUB TI/99     !
130 !                    !
140 !+-------------------+
150 !SUBPROGRAMA PARA REEM
160 !PLAZAR LOS JOYSTICKS !
170 !POR EL TECLADO SIN MO!
180 !DIFICAR EL PROGRAMA  !
190 !PRINCIPAL.           !
200 !+-------------------+
210 !        A=1          !
220 ! E=ARRIBA  X=ABAJO   !
230 !S=IZQUIERDA D=DERECHA!
240 !     Q=DISPARADOR    !
250 !+-------------------+
260 !        A=2          !
270 ! I=ARRIBA  M=ABAJO   !
280 !J=IZQUIERDA K=DERECHA!
290 !     Y=DISPARADOR    !
300 !+-------------------+
310 !
320 !
32000 SUB JOYST(A,X,Y):: CALL KEY(A,K,S):: IF K=2 THEN X=-4 :: Y=0 ELSE IF K=3 T
HEN X=4 :: Y=0 ELSE IF K=5 THEN X=0 :: Y=4 ELSE IF K=0 THEN X=0 :: Y=-4 ELSE X,Y
=0
32001 SUBEND
```

TRUCOS TAMPAS Y HALLAZGO



HACIENDO UN INPUT EN MEDIO DE LA PANTALLA

Muches veces al crear programas del tipo conversacional, nos encontramos con la necesidad de entrar datos en alguna posición particular de la pantalla y no necesariamente en la parte baja de la misma, como ocurre normalmente con la santencia INPUT. Pues bien, con un poco de picardía es perfectamente posible obtener este efecto como lo demuestra el siguiente programita:

```
10 INPUT AT 22,0: AT 0,0; "NOMBRE?"; N$
```

Las posibilidades de este efecto son infinitas, pero tiene un defecto fácil de descubrir. Una vez que hemos contestando al INPUT éste se borrará. Para evitarlo, podemos incluir esta línea que corregirá el dafecto:

```
20 PRINT AT 0,0;"NOMBRE?";N$
```

[illegible] sualizando la respuesta del INPUT tanto tiempo como querramos (pudiendo controlar desde programa el borrado, la nueva posición en pantalla de esa respuesta si la hubiera, etc.).

"METIENDO" VARIABLES EN UN INPUT

Continuando con los INPUTS, aquí tes ofrecemos un truquito para poder incluir una variable dentro de una sentencia INPUT. Para ello basta con usar el signo "+" para conectar un string con la variable que deseamos imprimir. De este forma, el ordenador construye primero el string antes de la impresión, y se consigue al efecto deseado.

Veamos un ejemplo práctico:

```
10 LET A$ = "PROGRAMA"
20 INPUT "NOMBRE DEL" + A$; B$
```

Naturalmente, podemos usar cualquier tipo de variable, normal o metriz, siempre que sea alfanumérica. Si queremos imprimir una variable numérica, tendremos que utilizar la función STR$ para con- [illegible]

CALIDAD DE IMPRESION

Para todos aquellos que poseemos una impresora ZX o TS-2040 o Alphacom 32, este truquito pueda resultar de utilidad. Tiene dos usos el primero es el de poner de manifiesto la calidad de impresión de nuestra impresora. El segundo es, sabiendo que nuestra impresora imprime O.K., testear la calidad del papel térmico a utilizar, dado que muchas veces nos encontramos con bobinas de papel antiguo o con "colitas" de rollos anteriormente utilizados.

El test es el siguiente:

```
10 LPRINT INVERSE 1; "(32 espacios)"
```

Con ésto obtendremos una linea impresa que nos mostrará la eficiencia de la impresión al momento de utilizarla.

EFECTO 1

Cuando estamos creando nuestro propio juego de aventuras, especialmente si se trate de uno del tipo "mercianitos", podemos echar mano de esta rutina en código máquina que [illegible] sonido bastante bueno. Para utilizarla basta con incluir la siguiente instrucción en toda aquella parte

del programa que sea necesario utilizarla:

RANDOMIZE USR 32400

La rutina es la que sigue y se puede teclear en cualquier parte del programa, por lo que no resulta molesto el incluirla.

```
20 CLEAR 32399
30 FOR A = 32400 TO 32425
40 READ B : POKE A,B
50 NEXT A
60 DATA 6,5,197,33,0,3,17,1,0,
229,205,181,3,225,17,15,0,
167,23,7,82,32,240,193,16,
233,201
```

EFECTO 2

Otro truco interesante consiste en la modificación de los colores del borde. Usualmente colocaríamos varias instrucciones BORDER seguidas:

10 BORDER 1 : BORDER 2 : BORDER 3 : BORDER 4 : BORDER 5 : BORDER 6 : BORDER 7 : PAUSE 0 BORDER 1 : BORDER 2 : PAUSE 1 : GO TO 10

o bien

```
10 BORDER 2
20 BORDER 5
30 GO TO 10
```

La instrucción PAUSE sirve para sincronizar el envío de las instrucciones BORDER con la frecuencia de barrido del televisor. Podemos probar que sucede eliminándola. Si durante la ejecución pulsamos una tecla, la velocidad de desplazamiento de las líneas horizontales que produce el programa se verá alterada. Esto se debe a que cuando el compulsador detecta el pulsado de una tecla, tiene que realizar varias operaciones, lo que disminuye la velocidad con que las instrucciones BORDER son enviadas al televisor, alterando por lo tanto la sincronización.

También podremos probar colocando estas tres líneas al final de otro programa más largo y veremos que la velocidad también cambia. La explicación a este fenómeno la debemos buscar en el intérprete BASIC que cuando ejecuta una instrucción GO TO (o GO SUB) busca la línea correspondiente desde el principio del programa. Esto significa que un GO TO 1 será mucho más rápido que un, por ejemplo, GO TO 9999 (salvo, por supuesto, que ésta sea la única línea del programa). De aquí desprendemos que es mucho más productivo colocar al principio todas aquellas rutinas que utilizamos con mayor frecuencia. El ahorro de tiempo puede ser considerable, sobre todo en programas largos.

Continuando con el BORDER, podemos experimentar con la instrucción OUT 254,A, dando a A un valor comprendido entre 0 y 255. Mejor aun el colocamos varios OUT seguidos en un bucle. Se consiguen efectos muy buenos.

Como ejemplo podemos probar:

```
10 FOR N = 1 TO 255
20 OUT 254, N
30 NEXT N
```

Pero esto no pueda sustituir al BORDER, dado que el efecto de un OUT es temporal. Para ver ésto, podemos teclear en modo directo: OUT 254, 2 y una vez ejecutado pulsamos una tecla y el color del borde vuelve el color original, siempre que éste no fuera rojo, claro.

PIRATAS II

Proponemos aquí un método alternativo que consideramos muy útil, dado que es muy difícil, por no decir imposible "reventar" un programa protegido por esta forma.

La cuestión está en impedir el funcionamiento de la tecla "BREAK" específicamente, de modo que cualquier intento de detener el programa, produce la destrucción del mismo.

Para ello nos servimos de que la computadora necesita saber, al ser pulsada una tecla, el contenido de ciertas posiciones de memoria, si cambiemos esas direcciones a otro lugar y las mandamos a otra parte, se producirá el "SISTEM CRASH" que buscamos.

Analizaremos el listado para saber cómo y porqué funciona:

Líneas 20-30:

Lo primero es fijar el RAMTOP del sistema, es decir, la última posición de la RAM antes de los gráficos definidos por el usuario.

La primera pregunta es porque usamos DEF FN en lugar de la sentencia LET, mucho más obvia; la razón está en que el resultado de la función la utilizamos en la línea 40, luego de ejecutar la sentencia CLEAR, que borra todas las variables, de este modo el computador no tendría idea de qué le estamos diciendo en la línea 40 (aparecería el famoso: VARIABLE NOT FOUND.).

LINEA 40:

Aquí la máquina evalua una expresión lógica, dando por resultado "1" si es verdadero o "0" si es falso; pretendemos asignar a la variable HPILA el valor de la variable PILA + 1. Como veremos mas adelante, INKEY$ efectivamente dará la cadena vacia cuando se ejecuta el programa. Así que la expresión es cierta y por lo tanto vale "1". La utilidad de este truco es muy grande, ya que nos permite, por ejemplo, bifurcar a varias líneas, según qué tecla hayamos pulsado.

LINEA 60:

Inhabilita la tecla "BREAK"

LINEA 90:

Restaura la tecla "BREAK" colocando los valores originales en las posiciones de memoria adecuadas.

LINEA 110:

Esto es una llamada a una rutina de la ROM que permite terminar un programa de une manera bastante curiosa; le dejemos al lector que realice sus propias experiencias. Para ver cómo funciona recomendamos incluir una línea con la siguiente

65 GO TO 65 + (INKEY$ = "a")

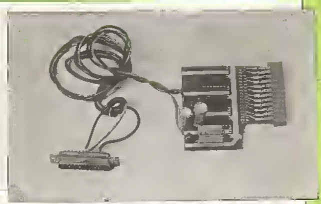
La pantalla permanecerá en blanco hasta que se pulse la tecla "a"; si pulsamos "BREAK" el programa se destruye. Recordemos que al hacer "RUN" la línea 40 contiene la cadena vacía, por lo que el resultado es verdadero.

En fin, aquí está expuesta la idea. No dudamos que nuestros lectores sabrán aprovecharla de muy buena gana.

```
10 REM **FIJA EL RAMTOP**
20 DEF FN R () = PEEK 23730 +
256 * PEEK 23731
30 CLEAR FN R ()
40 LET LPILA = FN R () - 3: LET HPI
LA = LPILA + (INKEY$ = "")
50 REM ** INHABILITA LA TECLA
"BREAK"**
60 POKE LPILA, 0: POKE HPILA, 0
70 REM ** RESTO DEL PROGRA-
MA **
80 REM ** RESTAURA "BREAK"**
90 POKE LPILA,3 : POKE HPILA,
19
100 RANDOMIZE USR 4710
```

COMO USAR LA INTERFASE SERIE DE LA C 64 (PRIMERA PARTE)

Aunque en los manuales de la C64 no diga nada, la máquina tiene incorporada una interface sene que se puede usar tacilmente desde el basic para enviar y recibir datos desde otra computadora o transmitir a una impresora serie.
Lo único qua hay que hacer es adaptar los niveles de salida que según las especificaciones del RS232 deben ser de -12V y + 12V pero que en la C64 son de 0y +5V (ver nota del número 3 sobre interface serie). Para lograr esto hay que colocar entre las lineas externas y la máquina unos adaptadoree de nivel como los archiconocidos 1488 y 1489. En el 1488 hay 4 amplificadores inversores que [illegible] en +12V y -12V respectivamente.



USER I/O
CONECTOR DE BORDE
12 LINEAS DOBLE FAZ

CONECTOR DIN 8 LINEAS

RECIBE
TRANSMITE
MASA

CONEXION MINIMA EN MODO 3 LINEAS

CARA DE COMPONENTES (Cara superior)

IC1 = 1488
IC2 = 1489

74LS04 PATA 14
+5V
1488
100 nF PATA 14
1488 PATA 7
1489 PATA 7
74LS04 PATA 7

DIODO RECTIFICADOR 200 mA

(Cara inferior)

100 µF
1488 PATA 14
500 µF 25 v
100 µF
1488 PATA 1

La microcamputadara Commadare 64 tiene incarparada una interfase serie que se puede usar fácilmente desde el Basic, para enviar y recibir datas desde atra máquina a transmitir a una impresara serie

El 1489 hace lo contrario o sea dada una señal de -12V en la entrada la translorma a +5V y de +12V a 0V. Entonces las señales que nosolros mandemos al exterior las haremos pasar por el conversor de vollaje 1488 y las que recibamos pasarán antes por el 1489.

ESQUEMA DE LA INTERFASE PROPUESTA:

Para su construcción (se puede conseguir armada) necesitaremos un conector de borde de 12 líneas doble laz que se conecta a la compuladora, una plaqueta de expenmentación y un conector para la salida. Normalmente se usa aquí un DB25, pero debido al alto coslo de estos, recomendamos una ficha din de 8 líneas (como la salida audio/video de la C64) o en caso que no usemos todas las líneas de salida una ficha din de 3 o 5 líneas. Ej. para conectar 2 C64 de la manera más simple posible haremos el siguiente cable:

C 64 — 1, 7, 8 ⟷ 1, 7, 8 — C64 (1 y 7 cruzados, 8 directo)

Para usar la interface desde el Basic usaremos los comandos open close, get#, input #, print# como los olros penféncos.

PATAS DE LOS IC

1 · 2 · 3 · 4 · 5 · 6 · 7 — 14 · 13 · 12 · 11 · 10 · 9 · 8

Lineas del Port de Usuario

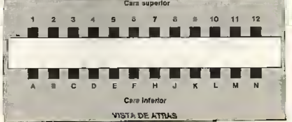
(6526 DEVICE # 2 Loc. $ DD00 - $DD0F)

PIN ID	6526 ID	DESCRIPCION	EIA	ABV	IN/OUT	MODOS
C	PB0	RECEIVED DATA	(BB)	Sin	IN	1*2
D	PB1	REQUEST TO SEND	(CA)	RTS	OUT	1*2
E	PB2	DATA TERMINAL READY	(CD)	DTR	OUT	1*2
F	PB3	RING INDICATOR	(CE)	RI	IN	3
H	PB4	RECEIVED LINE SIGNAL	(CF)	DCD	IN	2
J	PB5	UNASSIGNED	( )	XXX	IN	3
K	PB6	CLEAR TO SEND	(CB)	CTS	IN	2
L	PB7	DATA SET READY	(CC)	DSR	IN	2
B	FLAG 2	RECEIVED DATA	(BB)	Sin	IN	1 2
M	PA2	TRANSMITTED DATA	(BA)	Sout	OUT	1 2
A	GND	PROTECTIVE GROUND	(AA)	GND		1 2
N	GND	SIGNAL GROUND	(AB)	GND		1 2 3

MODOS:
1) 3-LINE INTERFACE (Sin, Sout, GND)
2) X-LINE INTERFACE
3) Utilizable por el usuario (no usado)
* Estas líneas permanecen allas en MODO 3-LINE

Registro de Estado (ST) del RS-232.

```
(7) (6) (5) (4) (3) (2) (1) (0)  (Machine Lang. — RSSTAT
 :   :   :   :   :   :   :   ' — PARITY ERROR BIT
 :   :   :   :   :   :   : ————FRAMING ERROR BIT
 :   :   :   :   :   : ————————RECEIVER BUFFER OVERRUN BIT
 :   :   :   :   : ————————————RECEIVER BUFFER — EMPTY
                                (USE TO TEST AFTER A GET)
 :   :   :   : ————————————————CTS SIGNAL MISSING BIT
 :   :   : ————————————————————UNUSED BIT
 :   : ————————————————————————DSR SIGNAL MISSING BIT
 : ————————————————————————————BREAK DETECTED BIT
```



VISTA DE ATRAS

COMO USAR LA INTERFASE SERIE DE LA C64

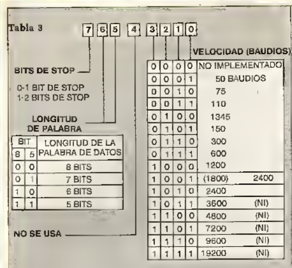


Tabla 3 — 7 6 5 4 3 2 1 0

BITS DE STOP
0-1 BIT DE STOP
1-2 BITS DE STOP

LONGITUD DE PALABRA

BIT		LONGITUD DE LA PALABRA DE DATOS
8	5	
0	0	8 BITS
0	1	7 BITS
1	0	6 BITS
1	1	5 BITS

NO SE USA

				VELOCIDAD (BAUDIOS)	
0	0	0	0	NO IMPLEMENTADO	
0	0	0	1	50 BAUDIOS	
0	0	1	0	75	
0	0	1	1	110	
0	1	0	0	1345	
0	1	0	1	150	
0	1	1	0	300	
0	1	1	1	600	
1	0	0	0	1200	
1	0	0	1	(1800)	2400
1	0	1	0	2400	
1	0	1	1	3600	(NI)
1	1	0	0	4800	(NI)
1	1	0	1	7200	(NI)
1	1	1	0	9600	(NI)
1	1	1	1	19200	(NI)

MAPA DE REGISTROS DE CONTROL.

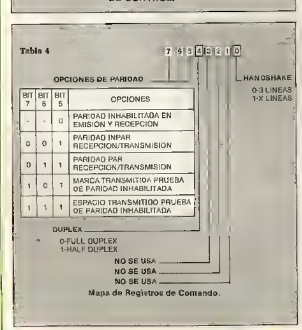


Tabla 4 — 7 6 5 4 3 2 1 0

OPCIONES DE PARIOAD

HANDSHAKE
0-3 LINEAS
1-X LINEAS

BIT 7	BIT 6	BIT 5	OPCIONES
-	-	0	PARIOAD INHABILITADA EN EMISION Y RECEPCION
0	0	1	PARIDAD INPAR RECEPCION/TRANSMISION
0	1	1	PARIOAD PAR RECEPCION/TRANSMISION
1	0	1	MARCA TRANSMITIOA PRUEBA DE PARIDAD INHABILITADA
1	1	1	ESPACIO TRANSMITIDO PRUEBA DE PARIDAD INHABILITADA

DUPLEX
0-FULL DUPLEX
1-HALF DUPLEX

NO SE USA
NO SE USA
NO SE USA

Mapa de Registros de Comando.

La instrucción open requiere el siguiente formato.
Open Lfn, 2,0,chr$ (nro. de control) +chr$ (nro. de comando + 2 bytes oplativos
Lfn indica el número de fichero lógico entre 1 y 255. Si es mayor que 127 después de cada relum transmitirá un line feed (chr$ (10))
El nro. de control entre 0 y 255 se calcula según la Tabla 3

Ejemplo: para transmitir a 1200 beu dios con longitud de palabra de 8 bits y 2 bits de stop el número de control deberá ser 128+8
Para transmitir 7 bits con 1 bit de stop a 300 band deberá ser 32+6
Si los bits 0 al 3 eslan en 0 la velocidad se elegirá según una fórmula en la que intervienen los bytes optalivos.
El número de comando se elegirá según la Tabla 4.
El modo 3-líneas significa qne solo estarán activas las líneas de transmisión recepción y masa, o sea que no se utilizarán líneas de handshaking. En este modo existe el peligro de la pérdida de caracteres si no se procesan con la suficiente rapidez ya sea en el receptor como en el transmisor pero es siempre conveniente probar primero asi por la simplicidad del conexionado.
Ej. si no usamos paridad y lo conectamos en modo 3-líneas el número de comando debe ser 0
NOTAS:
Para controlar los problemas de transmisión existe la variable reservada ST que toma distintos valores según el eslado de la comnnicación (ver la Tabla 2.)
En nn programa la apertura del canal serie debe ser ejecutada antes de delinir variables o tablas porqne las borra
Para recibir se puede hacer con la instrucción get# Lfn. (variable string)
Para transmitir con print# Lfn, (lista de variables)
La transmisión y recepción serie está bastante bien explicada en las guias de referencia de la C64. Aunque a primera vista pueda parecer complicado experimentando un poco se puede ser un experto en "hacer hablar a las computadoras"
En el próximo número daremos detalles de la conexión e impresoras y otras computadoras del mercado.

MARCELO O. MARTINEZ



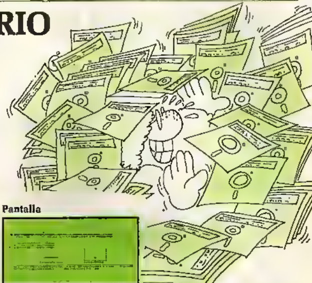
DIRECTORIO



COMP: C 64
CONF. 64 K
CLAS: UTI

Este utilitario tiene por finalidad hacer más rápida la lectura del Directorio que figura en el diskette y permite elegir el programa que se quiere usar. Ademas otras opciones como:

1) Lee el nombre del disco que se colocó en el drive.
2) Numera la cantidad de juegos que hay en dicho diskette.
3) La tecla F1 pasa a la página siguiente en caso que sea muy extenso el Directorio.
4) La tecla F3 es le secuencia para que en forma directa cargue en la memoria la opción elegida,
5) La tecla F5 situa al programa "DIRECTORIO" en la primera parte, ofreciendo la posibilidad de leer otro diskette en caso de buscar otros programas.



Pantalla

```
1 REM [illegible]
2 REM
3 REM        DIRECTORIO
4 REM           POR
5 REM      PANCHO MICHLE
6 REM
7 REM               1989
9 REM [illegible]
10 PRINT [illegible] 53281,0
20 REM [illegible]
25 REM    PRESENTACION.
30 REM [illegible]
33 REM
35 CLR:HISTORE
37 PRINT"
40 PRINT"  DIRECTORIO
43 PRINT"
45 PRINT"  PROGRAMADOR PANCHO MICHLE
47 PRINT"
50 PRINT
55 PRINT
57 PRINT"  INTRODUZCA
58 PRINT"  EL DISCO
60 PRINT"  EN SU DRIVE
[illegible]
80 PRINT"  DRIVE        DISK
[illegible]
87 PRINT"  PRESIONE UNA TECLA PARA CONTINUAR
88 PRINT"  ESPACIO PARA TERMINAR
[illegible]
91 GET A$:IF A$="" THEN GOTO 91
92 IF A$=" " THEN PRINT"[illegible]":END
93 FOR A=1 TO 10
94 PRINT"  [illegible]
95 PRINT"[illegible]
96 FOR I=1 TO [illegible]:NEXT I
97 NEXT A
98 PRINT [illegible] "DRIVE TRABAJANDO"
99 DIM F$(200)
100 REM [illegible]
105 REM    [illegible] ML ROUTINE
110 REM [illegible]
115 REM
120 REM
125 FOR X=828 TO 958
130 READ Z:POKE X,Z
135 CS=CS+Z:NEXT
140 IF CS<>10432 THEN GOTO 145
142 GOTO 150
145 PRINT"ERROR - REVISE SU DISCO DE DATA":END
150 REM [illegible]
155 REM    READ DIRECTORY
160 REM [illegible]
170 REM
180 REM
185 C=0
190 OPEN 15,8,15
200 PRINT#15,"I":[illegible]:GOSUB 290
210 OPEN 1,8,0,"$"+D$:GOSUB 290
220 N$=" [illegible] ":SYS 828
230 IF ST THEN CLOSE1:CLOSE15:GOTO300
240 I$=" [illegible] ":SYS 828
250 IF [illegible] THEN C=C+1:F$(C)=[illegible]LEFT$(F$,16)
260 GOTO 230
270
280 REM [illegible]
285 REM    DISK ERROR ROUTINE
290 REM [illegible]
[illegible] IF [illegible] THEN RETURN
[illegible]
310 CLOSE [illegible]
320 REM [illegible]
325 REM    DISPLAY ROUTINE
330 REM [illegible]
340 IF F>0 OR F<C THEN [illegible] THEN F=40:GOTO [illegible]
345 [illegible] THEN [illegible]
350 REM [illegible]
355 REM    DISPLAY MENU ROUTINE
360 REM [illegible]
[illegible]
390 FOR [illegible] TO S+19
395 [illegible]
[illegible] IF [illegible] THEN GOTO 420
400 IF [illegible] THEN GOTO 420
410 PRINT [illegible]
415 NEXT
420 PRINT TAB(20)"[illegible]
421 PRINT TAB(20)"
422 PRINT TAB(20)"
423 PRINT TAB(20)"
424 PRINT TAB(20)"
425 PRINT TAB(20)"
426 PRINT TAB(20)"
427 PRINT TAB(20)"
428 PRINT TAB(20)"
429 PRINT TAB(20)"  DISK
430 PRINT TAB(20)"
431 PRINT TAB(20)"
432 PRINT TAB(20)"
[illegible]
[illegible] GET [illegible]
[illegible] IF [illegible] THEN GOTO 340
[illegible] IF [illegible] THEN GOTO 500
[illegible] IF [illegible] THEN GOTO 1
[illegible] GOTO [illegible]
[illegible]
[illegible] PRINT"  POR FAVOR ELIJA
[illegible] PRINT"  EL NUMERO DE [illegible]
[illegible] INPUT"  ELEGIDO"
545 G=VAL(G$):IF G [illegible] THEN [illegible]
550 REM --CHAIN ROUTINE--
[illegible]
[illegible] LEFT$(F$ [illegible]
580 LOAD F$,8
[illegible]
590 PRINT [illegible]
595 PRINT"  PRESIONE [illegible]
600 PRINT"[illegible]
850 REM [illegible]
860 REM    ML ROUTINE [illegible]
870 REM [illegible]
873 REM
875 REM
877 REM
880 DATA [illegible]
890 DATA [illegible]
900 DATA [illegible]
910 DATA [illegible]
920 DATA [illegible]
930 DATA [illegible]
940 DATA [illegible]
950 DATA 96
```

CONNET FOUR



COMP: C64
CONF: 64K
CLAS: ENT

Juguemos contra nuestro adversario y derrotémoslo alineando 4 piezas antes que él.
Las instrucciones son claras y aparecen en pantalla.



```
T REM ADAPTADO POR E.O.S. Y F.J.M.
TO ROM=190 POKE53280,1 POKE53281,1 REM          EOR COMMODORE 64
100 GG$="[illegible]"
120 DIMA 7: A  [illegible] 2) SO=49 OPEN1,0,0 M$="[illegible]"
130 PRINT"NECESITA INSTRUCCIONES (Y/N)"
140 GETU$ IFU$=""THEN140
150 IFU$<>"Y"ANDU$<>"N"THEN140
160 IFU$="Y"THEN700
170 FORI=1TO2 PRINT"JUGADOR "J"(MAX. 6 LETRAS)?", INPUT#1,N$(J)
180 IFLEN(N$(J))>6THENN$(J)=MID$(N$(J),1,6)
190 NEXT
200 PRINT"[illegible]" FORZ=1TO15
210 PRINT" XX   XX   XX   XX   XX   XX   XX"
220 NEXT
230 FORZ=1TO2:PRINT" [illegible]" NEXT
240 PRINT"  1   2   3   4   5   6   7"
250 FORF=1TO7 READA(E) NEXT
260 DATA1187,1267,1347,1427,1507,1507,1667
270 FORF=1TO7 READB(E) NEXT
280 DATA1027,1032,1037,1042,1047,1052,1057
290 PRINTGG$ N$(1) [illegible] "N$(2)"[illegible]"BL
300 GETR$ IFR$=""THEN300
310 IFVAL(R$)=0ORVAL(R$)>7THEN300
320 P=VAL(R$)
330 IFAX(P)=7THEN300
340 IFN=1GOTHENT=102 POKE1989,81 POKE56293,0 POKE2012,32 GOTO360
350 N=160 POKE2012,81 POKE56294,0 POKE1989,32
360 GOSUB400  AX(P)=AX(P)+T GOTO430
370 SO=SO-T POKEROM,0
380 IFSO=0THENPRINT"[illegible]"SPC(15)"NO GANA !!" FORQQ=1TO1500 NEXT GOTO640
390 GOTO300
400 FORW=B(P)TOA(7-AX(P))+54STEP40
410 FORO=WTOW+2 POKEO+40,N POKEO-40,32 NEXT POKEA,N/2:FORT=1TO5 NEXT POKEA,0
420 NEXT RETURN
430 B%(P,AX(P))=N
440 FORH=AX(P)TO1STEP-1 IFB%(P,H)<>NTHEN460
450 CO=CO+1 NEXT
460 IFCO>=4THEN570
470 FORH=1TO3 IFB%(P+H,AX(P))=NTHENNEXT
480 FORI=1TO3 IFB%((P-I),AX(P))=NTHENNEXT
490 IFI+H>4THEN570
500 FORH=1TO3 IFB%(P+H,AX(P)+H)=NTHENNEXT
510 FORI=1TO3 IFB%(P-I,AX(P)-I)=NTHENNEXT
520 IFI+H>4THEN570
530 FORH=1TO3 IFB%(P-H,AX(P)+H)=NTHENNEXT
540 FORI=1TO3 IFB%(P+I,AX(P)-I)=NTHENNEXT
550 IFI+H>4THEN570
560 CO=0 GOTO370
570 FORL=1TO3 FORQ=0TO255STEP5 POKER,0 NEXT POKER,0 NEXT
580 FORW=1874TO1114STEP40
590 FORO=W[illegible] POKEO N NEXT NEXT
600 PRINT"[illegible]"SPC(14)"ES EL GANADOR !!"
610 FORPE=1TO1000 NEXT
620 IFT=102THENBL=BL+T GOTO640
630 WH=WH+1
640 RESTORE FORX=1TO7 AX(X)=0 NEXT SO=49 CO=0
650 FORX=1TO7 FORY=1TO7 B%(X,Y)=0 NEXT NEXT
660 PRINT"[illegible]", FORTY=1TO3 PRINT"                    " NEXT
670 PRINT FORZ=1TO15
680 PRINT"  XX   XX   XX   XX   XX   XX   XX"
690 NEXT FORM 0 N=102 POKE1989,81 POKE2012,32 GOTO250
700 PRINT"[illegible]"
710 POKE53272,23 FORE=1TO22 PRINT"[illegible]" NEXT
720 FORT=1TO2000 NEXT PRINT"[illegible]"
730 PRINT" EL OBJETIVO DEL JUEGO ES ENCOLUMNAR CUATRO PIEZAS ",
740 PRINT"LO PUEDE HACER EN HORIZONTAL EN VERTICAL O EN DIAGONAL"
750 PRINT" PARA MOVER UD. Y SU OPONENTE IRAN TURNOS COLOCANDO SUS PIEZAS"
760 PRINT" DE COLOR EN UNO DE LAS SIETE COLUMNAS NUMERADAS"
770 PRINT" HARA ESTO TIPEANDO EL NUMERO DE COLUMNA QUE HA ELEGIDO PARA COLOCAR"
780 PRINT"SU PIEZA . EL JUEGO TERMINA CUANDO UNO DE LOS DOS JUGADORES HAYA COLO"
790 PRINT"CADO CUATRO DE SUS PIEZAS CONSECUTIVAMENTE"
810 PRINTSPC(7)"PRESIONE UNA TECLA",
820 GETU$ IFU$<>""THENPOKE53272,21 GOTO170
830 GOTO820
```

CHEQUEAR JOYSTICK



COMP: C64
CONF: 64K
CLAS: UTI

Este programa permite el chequeo de un mando de juego (joystick), a través del port. Nº 2 de la Commodore, y verifica su funcionamiento.




```
[illegible]
```

AJEDREZ

COMP: C64
CONF: 64K
CLAS: ENT.

Probemos nuestra habilidad y estrategia jugando con la Commodore al Ajedraz.

```
10 REM CHESS BOARD 1.5
20 GOSUB20000
30 PRINT"■", POKE53280,1 POKE53281,1
1000 GOSUB 10
1100 GOSUB5000
1200 GOSUB11000
1300 L$=LEFT$(I$,2) GOSUB12000
1310 IFF<>0THEN1400
1320 PRINT PRINT"LO LAMENTO NO HAY PIEZA",
1330 FORI=1TO1000 NEXT GOTO1200
1400 D(F)=Z(4)*(I 3)
1405 FORI=1TO10
1410 F$=LEFT$(I$,2) D=D(F)
1420 GOSUB6000
1430 F$=RIGHT$(T$,2) D=0
1440 GOSUB6000
1450 F$=LEFT$(I$,2) D=0
1460 GOSUB6000
1470 F$=RIGHT$(T$,2) D=D(F)
1480 GOSUB6000
1490 NEXT
1500 GOTO1200
7999 END
6000 REM SETUP A PIECE
6040 I=ABS(D) IFD<0THENI=I+9
6070 Q$=P$(I)
6080 X=ASC(F$)-64:Y=VAL(RIGHT$(F$,1))
6100 G=X+Y
6120 R$=""
6140 IFG/2<>INT(G/2)THENR$="■"
6150 PRINTLEFT$(D$,4+2*(9-Y)) SPC(4+(4*(X-1)))R$;Q$, "■"
6200 RETURN
6500 REM SETUP BOARD
6520 FORI=1TO34
6540 IFD$(I)=" "THEN6600
6550 F$=B$(I) D=D(I)
6570 GOSUB6000
6600 NEXT
6650 RETURN
9000 PRINT"■            AJEDREZ PARA DOS"
9003 PRINT"     A   B   C   D   E   F   G   H "
9005 PRINT"    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━"
9010 FORI=1TO4
9020 FORJ=1TO2
9025 A$=" "
9030 IFJ=2THENA$=CHR$(58-2*I)
9040 PRINT" "A$" ■"
9050 FORK=1TO4 PRINT"■    ■     " NEXT
9060 PRINT"■ "A$
9070 NEXT
9080 FORJ=1TO2
9085 A$=" "
9090 IFJ=2THENA$=CHR$(57-2*I)
9100 PRINT" "A$" ■",
9110 FORK=1TO4 PRINT"     ■    ■", NEXT
9120 PRINT"■ "A$
9130 NEXT
9140 NEXT
9150 PRINT"    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━"
9155 PRINT"     A   B   C   D   E   F   G   H "
9160 RETURN
10000 REM WAITING GET
10010 PRINT"■F"
10020 FORI=1TO20
10030 GETS$ IFS$<>""THEN10100
10040 NEXT
10050 PRINT" ■",
10060 FORI=1TO50
10070 GETS$ IFS$<>""THEN10100
10080 NEXT
10090 GOTO10010
10100 PRINT" ■"
10110 RETURN
11000 REM GETS A MOVE
11010 PRINTD$,E$ PRINTE$ O"ENTRE SU JUEGO "
11015 I$=""
11020 FORJ=1TO4
11030 GOSUB10000
11040 S=ASC(S$)
11050 IF(J=1ORJ=3)AND(S<65ORS>72)THEN11200
11060 IF(J=2ORJ=4)AND(S<49ORS>56)THEN11200
11080 PRINTS$, I$=I$+S$
11090 NEXT
11100 PRINT"     (PRESIONE RETURN)"
11110 GOSUB10000
11120 IFS$<>CHR$(13)THEN11010
11130 RETURN
11200 PRINT PRINT"ERROR    PRUEBE DE NUEVO
11210 FORI=1TO1000 NEXT
11220 GOTO11010
12000 REM CHECK PIECE
12010 F=0
12020 FORI=1TO34
12030 IFB$(I)=L$THENF=I
12040 NEXT
12050 RETURN
20000 DIM P$(18),D(34),B$(34)
20010 FORI=0TO18 READ P$(I) NEXT
20020 FORI=1TO34 READ D(I),B$(I) NEXT
20030 D$="■" FORI=1TO21 D$=D$+"■" NEXT
20040 FORI=1TO39 E$=E$+" " NEXT
20050 FORI=1TO39 E$=E$+"■" NEXT
21000 RETURN
30000 REM PIECE DATA
30005 DATA"     ■■■■■    "
30010 DATA"  ▫  ■■■■■    "
30020 DATA"  ─  ■■■■■    "
30030 DATA"| |  ■■■■■  ┘"
30040 DATA""
30050 DATA"└┘  ■■■■■  "
30060 DATA""
30070 DATA
30080 DATA"└┘  ■■■■■  "
30090 DATA"└┘  ■■■■■  "
30110 DATA"  ╭  ■■■■■  "
30120 DATA"  ╱  ■■■■■  "
30130 DATA" |-| ■■■■■ | "
30140 DATA""
30150 DATA" |-| ■■■■■ ┌┐"
30160 DATA""
30170 DATA""
30180 DATA" \/  ■■■■■ ┌┐"
30190 DATA" HH  ■■■■■ ┌┐"
31000 REM BOARD DATA
31010 DATA 1,A7, 1,B7, 1,C7, 1,D7, 1,E7, 1,F7, 1,G7, 1,H7
31020 DATA 2,B8, 2,G8
31030 DATA 3,C8, 3,F8
31040 DATA 5,A8, 5,H8
31050 DATA 8,D8, 8,""
31060 DATA 9,E8
31110 DATA -1,A2,-1,B2,-1,C2,-1,D2,-1,E2,-1,F2,-1,G2,-1,H2
31120 DATA -2,B1,-2,G1
31130 DATA -3,C1,-3,F1
31150 DATA -5,A1,-5,H1
31160 DATA -8,D1,-8,""
31190 DATA -9,E1
```

D E B U G G I N G

En adelante en esta sección volcaremos todas aquellas dificultades que nuestros lectores nos hagan llegar, respecto a las notas o programas de K-64. También daremos todas las explicaciones necesarias para subsanar errores o "BUGS" que se nos pudiesen escapar, (que como saben los dedicados a la informática, es "inevitable").

A fines de la segunda guerra mundial, para el cálculo de trayectorias de proyectiles, se utilizaban enormes computadoras que funcionaban en base a reles. (interruptores electromganéticos) En uno de esos días, una gran máquina se negaba a funcionar. Había un defecto en su funcionamiento. Como aún no existía el "soft", sino solo el "hard", donde el programa consistía no en conexiones lógicas en una memoria, sino en conexiones físicas, por medio de innumerables cables formando enormes marañas, el error debía estar necesariamente en un problema mecánico. Las necesidades de operación crecían, y las presiones también. Hasta que por fin encontraron la falla: una enorme polilla estaba atrapada entre los contactos de uno de los reles.

De allí que el término inglés "BUG", (insecto), se refiera a los errores como fidos en la programación. La descubridora de este intruso, fue nada menos que Grace HOOPER, creadora del primer lenguaje de programación (el A-0) y colaboradora del equipo que desarrolló el popular COBOL. Nuestros lectores, de aguda vista y sentido analítico, nos hacen notar los "BUGS" que aparecen en nuestras páginas. Les agradecemos mucho su colaboración.

ACLARACIONES SOBRE COMPATIBILIDAD

Es necesario aclarecer algunos puntos respecto a la compatibilidad de ciertos equipos. Entre las computadoras Sinclair o Czerweny en sus modelos 1000 / 1500 y los equipos Microdigital en sus modelos TK-83 / 85, la compatibilidad en BASIC es total, excepctuando las funciones de carga rápida del TK-85). En los programas en lenguaje máquina, son raros los casos (pero los hay) en que los programas no son permutables.

En cuanto a los modelos CZ-2000, TK-90 X y TS-2068, son absolutamente compatibles EN LAS FUNCIONES BASIC COMUNES. En lenguaje Assembler, los modelos CZ-2000 y TK-90 X se muestran mucho más adaptables entre sí, en cambio es mucho más factible que se produzca un error con programas de CZ-2000 ó TK-90 X corridos en una TS-2068 y viceversa.

ORGANO PERDIDO (Nº 2, Pág. 33)

En el programa órgano, para ZX Spectrum, debemos afirmar que lo que allí aparece es el listado del programa... menos un tercio del mismo, que hasta el día de hoy desconocemos su paradero Vaya uno a saber en que insospechado rincón se encuentre en este momento

Pero no os aflijáis, osados computómanos, si dentro de la diagramación del presente número ya han visto, el programa MUSICO, una excelente herramienta para

DEBUGGING

dar rienda suelta a nuestra inspiración.

VARIABLES FANTASMA
(Nº 3, Pág. 9)

Aquí está por fin, el listado de las variables "fantasma" del programa SUPERCALC.

1) Listado de variables numéricas:

Q = 20	W = 100	E = 0
R = 200	T = 16426	Y = 39
U = 1	O = 13	P = 1000
A = 0	S = 16425	D = 0
F = 0	G = 0	H = 0
J = 74	K = 19	L = 5
Z = 39	X = 256	C = 0
V = 11	B = 34	M = 1

2) Listado de variables alfanuméricas:

SS = " "
WS = " 16 espacios"
PS = " "
JS = " "
BS = " "

Una vez tecleado el programa, se deben introducir las variables de los listados 1) y 2), sin número de línea.

Finalizada la tarea, inicializar con GOTO U. Jamás arranquemos el programa con RUN y nunca utilicemos la instrucción CLEAR.

Si se dispone de una memoria superior a 2 K, estas variables pueden ser ingresadas como líneas de programa a partir de la línea 9000. En este caso, la línea 9010 GOTO U debe modificarse por 9030 GOTO U, si es que la última línea de programa es la 9030.

REEMPLAZO EN EL SIMULADOR
(Nº 3, Pág. 17)

En el programa simulador de vuelo, en el programa generador de REM:

3 PRINT "CANTIDAD DE BYTES?"

Debe reemplazarse por:

3 PRINT "CUANTOS BYTES DE LONGITUD? "

Es importante respetar el número de caracteres de esta línea, así como el espacio final antes de cerrar comillas.

Debajo del cuadrito de la explicación en Assembler, siempre en página 17, se lee lo siguiente

LINEA 11: POKE A + 1,61

Debe leerse:

LINEA 11 : POKE A + 1,61

Esta línea es exactamente la misma línea 11 del programa generador de REM.

CALENDARIO Y RULETA
(Nº 5, Pág. 10 y 42)

En el programa "Calendario", las tres últimas líneas son las siguientes.

350 INPUT A$
360 CLS
370 IF A$ = "S" THEN GOTO 10

Mientras que en el programa "Ruleta Rusa", la última línea es la siguiente.

9120 GOTO 1



CONCURSO TRIMESTRAL

NUMEROS MAGICOS

Comp: TK 85/CZ - 1500/
Conf: 16 K
Clas: Entretenimiento
Autor: Raúl Daniel Vera



Este programa, de entretenimiento, consiste en un diálogo operador/máquina, donde las únicas contestaciones posibles son "si" o "no". De tal manera que resulta fácil seguir el modo conversacional que se plantea. Por este razón no es necesaria una descripción de su uso y otros detalles.

Las subrutinas de scroll horizontal son particularmente vistosas. Se destaca también el uso del modo FAST en aquellos pasajes de presentación de tableros de números (debido a lo cansador que resultaría esperar a que se escriban todos ellos).

```
N U M E R O S   M A G I C O S

*POR EAVOR PERMITPME CONOCERLO..

ESCRIBA EU NOMBPE  JULIO

TAMBIEN SU REELLIDO  PEREZ

GRACIAS JULIO.....

FECHA  T  DE  AGOSTO  DE  T98E


*USTED NO CREERA LO QUE A CONTI-
NUACION EXACTAMENTAEE ...*

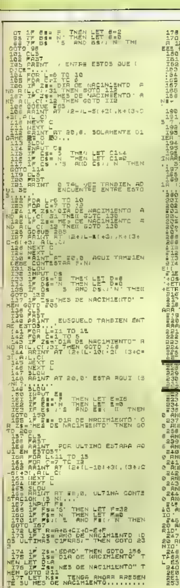

BIEN AHORA LE MOSTERE UN CONJUN
TO DE NUMEROS...


        2   E   7   9  11   1
       T3  TS  17  19  21  23
       ;E  27  29  FT  33  35
       37  39  4T  43  45  47
       49  51  E3  55  E7  E9


CONTESTEME (E/N) SU EDAD
EETA ENTPE EETOF NUMEROS?...


   1 CLS
   2 REM [illegible]
  20 CLS
  21 PRINT ************************
**************
  22 PRINT ***** AUNQUE UD.NO L
CEEA, YO LEADIUTNARE SU EDAD ,
-MUCHO MAS,...
  23 PRINT ************************
*************
  24 PRINT AT TE,0 ST QUIERE PR
OBAR TOQUE CUALQUIERTECLA SALVO
BREAK ***********
  25 IF INKEY$= "" THEN GOTO 25
  26 LET K$= ***- LO UNICO QUE D
EBE HACER ES CONTESTAR S: O NO..
..
  27 GOSUB 228
  28 PRINT AT T3,0, *** D E   A
C U E R D O ...
  29 PRINT AT 21,0 CONTESTE (S
N)
  30 IF INKEY$="S" OR INKEY$="SI"
THEN GOTO 38
  31 IF INKEY$="N" OR INKEY$="NO"
THEN GOTO 33
  32 GOTO 30
  33 LET K$="****** ES UD, UN
VIEJO ARCAICOCONTEMPORANEO DE MA
TUSALEM...
  34 GOSUB 22E
  35 PRINT AT 10 0 NO LE DESPNT
ME VENGE ......
  36 FOR T=T TO 300
  37 NEXT T
  38 CLS
  40 PRINT  N U M E R O S   M
A G I C O S
  41 PRINT
  42 LET K$= *POR EAVOR PERMITAM
E CONOCERLO..
  43 GOSUB 227
  44 PRINT AT 9 0, ESCRIBA SU NO
MBRE  ",
  45 INPUT N$
  46 PRINT N$
  47 PRINT AT 11,0 TAMBIEN SU A
PELLIDO "
  48 INPUT O$
  49 PRINT O$
  50 PRINT AT 13 2 GRACIAS  N$
" ....
  51 PRINT AT TE 0 FECHA ACTUAL
?  (EN NUMEROS)
  52 PRINT AT 21,0, PRIMERO EL D
IA (1 E 31) DE HOY,.'
  53 INPUT H
  54 IF H<1 OR H>31 THEN GOTO 53
  55 PRINT AT 2T 0 AHORA EL MES
(T A 12) ACTUAL...  AT 15 14,H

  56 INPUT U
  57 IF U 1 OR U 12 THEN GOTO 56
  58 PRINT AT E, 0 AHORA ESCRIBE
A EL ANO COMPLETO   AT TE TT U
  59 INPUT U
  60 IF U 1982 OR ... 2000 THEN GO
TO 59
  61 PRINT AT 15 0 FECHA   H
DE  MES ...
  62 PRINT AT E0 E *USTED NO CE
EERA LO QUE A CONTINUACION EXPE
RIMENTAES ...
  63 FOR E=T TO 200
  64 NEXT :
  65 LET ZE= 1040
  66 CLS
  67 PRINT  BIEN AHORA LE MOSTEE
ZE UN CONJUNTO DE NUMEROS...'
  70 FOR L=1 TO 6
  71 FOR C=T TO 6
  72 IF Z$="DIA DE NACIMIENTO" A
ND A(L C) 31 THEN GOTO 7E
  73 IF Z$="MES DE NACIMIENTO" A
ND A(L C) 12 THEN GOTO 7F
  74 GOSUB 2FF
  75 PRINT AT 2*L+3 K+(3*C+3)
A(L C)
  76 NEXT C
  77 NEXT L
  78 PRINT AT 13 0 CONTESTEME (
S/N) ... Z$
  79 PRINT ESTA ENTRE ESTOS NUM
EROS'...
  80 INPUT R$
  81 IF R$= S THEN LET R=1
  82 IF R$= N THEN LET R=0
  90 IF R$ S AND R$ N THEN
GOTO 80
  8E CLS
  85 PRINT 'TAL VEZ ENTRE ESTOS.
......
  87 FOR L=1 TO 1
  88 FOR C=T TO T2
  89 IF Z$= DIA DE NACIMIENTO  A
ND A(L C) 31 THEN GOTO 9E
  90 IF Z$= MES DE NACIMIENTO' A
ND A(L C) T2 THEN GOTO 9E
  91 GOSUB 260
  92 PRINT AT (2*L+3) K+(3*(C-6)
+3),A(L,C)
  93 NEXT C
  94 NEXT L
  95 PRINT AT 20,0 CONTESTE NUE
VAMENTE (E/N)...
  96 INPUT BE
```



```
 97 IF B$="S" THEN LET B=2
 98 IF B$="N" THEN LET B=0
 99 IF B$<>"S" AND B$<>"N" THEN
GOTO 96
101 CLS
102 FAST
103 PRINT "¿ ENTRE ESTOS QUE (
DICE?..."
104 FOR L=6 TO 10
105 FOR C=1 TO 6
106 IF Z$="DIA DE NACIMIENTO" A
ND A(L,C)>31 THEN GOTO 112
107 IF Z$="MES DE NACIMIENTO" A
ND A(L,C)>12 THEN GOTO 112
108 GOSUB 285
109 PRINT AT (2+(L-5)*3),K+(3*C
+3),A(L,C)
110 NEXT C
111 NEXT L
112 PRINT AT 20,0;"SOLAMENTE DI
GAME [illegible] O [illegible]..."
113 SLOW
114 INPUT C$
115 IF C$="S" THEN LET C1=4
116 IF C$="N" THEN LET C1=0
117 IF C$<>"S" AND C$<>"N" THEN
GOTO 114
119 CLS
120 FAST
121 PRINT "O TAL VEZ TANBIEN AQ
UI SE       ENCUENTRE ENTRE ESTO
S....."
122 FOR L=6 TO 10
123 FOR C=7 TO 12
124 IF Z$="DIA DE NACIMIENTO" A
ND A(L,C)>31 THEN GOTO 130
125 IF Z$="MES DE NACIMIENTO" A
ND A(L,C)>12 THEN GOTO 130
126 GOSUB 285
127 PRINT AT (2+(L-5)*3),K+(3*(
C-6)+3),A(L,C)
128 NEXT C
129 NEXT L
130 PRINT AT 20,0;"AQUI TAMBIEN
DEBE CONTESTAR (S/N)"
131 SLOW
132 INPUT D$
133 IF D$="S" THEN LET D=8
134 IF D$="N" THEN LET D=0
135 IF D$<>"S" AND D$<>"N" THEN
GOTO 132
137 IF Z$="MES DE NACIMIENTO" T
HEN GOTO 206
138 CLS
139 FAST
140 PRINT "BUSQUELO TAMBIEN ENT
RE ESTOS..."
141 FOR L=11 TO 15
142 FOR C=1 TO 6
143 IF Z$="DIA DE NACIMIENTO" A
ND A(L,C)>31 THEN GOTO 148
144 PRINT AT (2+(L-10)*3),(3*C+
3),A(L,C)
145 NEXT C
146 NEXT L
147 SLOW
148 PRINT AT 20,0;"ESTA AQUI (S
/N)?......"
149 SLOW
150 INPUT E$
151 IF E$="S" THEN LET E=16
152 IF E$="N" THEN LET E=0
153 IF E$<>"S" AND E$<>"N" THEN
GOTO 150
155 IF Z$="DIA DE NACIMIENTO" O
R Z$="MES DE NACIMIENTO" THEN GO
TO 206
157 CLS
158 FAST
159 PRINT "POR ULTIMO ESTARA AQ
UI EN ESTOS?"
160 FOR L=11 TO 15
161 FOR C=7 TO 12
162 PRINT AT (2+(L-10)*3),(3*(C
-6)+3),A(L,C)
163 NEXT C
164 NEXT L
165 SLOW
166 PRINT AT 20,0;"ULTIMA CONTE
STACION (S/N)..."
167 INPUT F$
168 IF F$="S" THEN LET F=32
169 IF F$="N" THEN LET F=0
170 IF F$<>"S" AND F$<>"N" THEN
GOTO 167
172 LET R=A+B+C1+D+E+F
173 IF Z$="ANO DE NACIMIENTO (D
OS ULTIMAS CIFRAS)" THEN GOTO 23
3
174 IF Z$="EDAD" THEN GOTO 186
175 IF Z$="DIA DE NACIMIENTO" T
HEN LET D=A
176 IF Z$="MES DE NACIMIENTO" T
HEN GOTO 211
177 LET K$="  TENGA AHORA PRESEN
TE SU MES DE NACIMIENTO........."
```

```
178 GOSUB 226
179 PRINT AT 20,0;"EL NUMERO (1
A 12) DEL MES DE SU NACIMIENTO D
EBE RETENER..."
180 FOR I=1 TO 100
181 NEXT I
182 LET Z$="MES DE NACIMIENTO"
183 CLS
184 GOTO 70
185 STOP
187 IF A=0 THEN GOTO 253
188 LET A=R
189 LET K$="****** SU EDAD   *
N*  ES DE  "+STR$ A+" ANOS......"
190 GOSUB 276
191 LET K$="* S E G U I N D
S ?..."
192 GOSUB 226
193 PRINT AT 20,0;"TOQUE CUALQU
IER TECLA >>"
194 IF INKEY$="" THEN GOTO 194
195 CLS
196 LET K$="***** AHORA LE ADIV
INARE SU FECHA DE NACIMIENTO....
.. SI ?..."
197 GOSUB 276
198 CLS
199 LET K$="COMENZEMOS CON EL D
IA (1 A 31) DE SU NACIMIENTO ..
....."
200 GOSUB 226
201 FOR I=1 TO 60
202 NEXT I
203 LET Z$="DIA DE NACIMIENTO"
204 CLS
205 GOTO 70
206 LET F=0
207 GOTO [illegible]
208 LET E=0
209 LET F=0
210 GOTO [illegible]
212 LET E=R
213 LET T=1985-F
214 LET T=7-(J/5 OR (H<6 AND J=
E))
215 LET K$="***** BIEN, "+N$+
" "+O$+", USTED HA NACIDO EL DIA
"+STR$ (D)+"  DE  "+M$(S)+"  DE
"+STR$ (T)+" . *****"
216 GOSUB 276
217 GOTO 236
218 LET K$="** GRACIAS, NO ES P
ARA TANTO ..."
219 GOSUB 225
220 PRINT AT 21,0;"* * (NEWLINE
PARA CONTINUAR) * *"
221 INPUT I$
222 CLS
223 GOTO 30
224 STOP
225 CLS
227 PRINT
228 PRINT
229 PRINT K$
230 PRINT
231 PRINT
232 RETURN
233 LET A=85-A
234 GOTO 189
236 IF (D>=21 AND S=1) OR (D<=1
9 AND S=2) THEN LET S$="ACUARIO"
237 IF (D>=21 AND S=2) OR (D<=2
0 AND S=3) THEN LET S$="PISCIS"
238 IF (D>=21 AND S=3) OR (D<=2
0 AND S=4) THEN LET S$="ARIES"
239 IF (D>=21 AND S=4) OR (D<=2
0 AND S=5) THEN LET S$="TAURO"
240 IF (D>=21 AND S=5) OR (D<=2
0 AND S=6) THEN LET S$="GEMINIS"
241 IF (D>=21 AND S=6) OR (D<=2
0 AND S=7) THEN LET S$="CANCER"
242 IF (D>=21 AND S=7) OR (D<=2
0 AND S=8) THEN LET S$="LEO"
243 IF (D>=21 AND S=8) OR (D<=2
0 AND S=9) THEN LET S$="VIRGO"
244 IF (D>=21 AND S=9) OR (D<=2
0 AND S=10) THEN LET S$="LIBRA"
245 IF (D>=21 AND S=10) OR (D<=2
0 AND S=11) THEN LET S$="ESCORP
IO"
246 IF (D>=21 AND S=11) OR (D<=
20 AND S=12) THEN LET S$="SAGIT
RIO"
247 IF (D>=21 AND S=12) OR (D<=
20 AND S=1) THEN LET S$="CAPRIC
NIO"
248 LET K$="... PERDONE, ME E
STABA OLVIDANDO, SU SIGNO ZODIACA
L ES  "+S$+" , NO ES ASI?...."
249 GOSUB 276
250 GOTO 218
251 STOP
253 LET K$="***** COMO USTED E
```

```
UNA PERSONA MAYOR DE 62 ANOS LE
RUEGO QUE TENGA PRESENTE LAS DO
S ULTIMAS CIFRAS DEL ANO DE SU N
ACIMIENTO.*****"
254 GOSUB 276
255 CLS
256 LET K$="***** CONTINUAMOS...
....."
257 GOSUB 226
258 FOR I=1 TO 50
259 NEXT I
260 LET Z$="ANO DE NACIMIENTO (
DOS ULTIMAS CIFRAS)"
261 CLS
262 GOTO 70
264 SAVE "MAGICOS"
265 GOTO 293
276 CLS
277 PRINT AT 0,0;"[illegible]"
278 PRINT AT [illegible]
279 LET J=LEN K$
280 LET C2=1
281 IF C2<31 THEN PRINT AT 2,(3
1-C2);K$( TO C2)
282 IF C2>=31 AND C2<=LEN K$ TH
EN PRINT AT 2,0;K$(C2-30 TO C2)
283 IF C2>=LEN K$ THEN LET K$=K
$+" "
284 IF LEN K$=J+32 THEN GOTO 28
7
285 LET C2=C2+1
286 GOTO 281
287 RETURN
289 IF A(L,C)<10 THEN LET K=1
290 IF A(L,C)>=10 THEN LET K=0
291 RETURN
293 CLS
294 FOR I=1 TO [illegible]
295 PRINT AT 0,0;"[illegible] NUMER
OS MAGICOS"
296 PRINT AT 10,0;"PROHIBIDO [1
DITAR EENTE[C1A RUN" AT 10,28
" AT 10,28;" ";AT 10,28;"RUN"
298 NEXT I
299 PRINT AT 12,0;"INTERPRETACIO
NES ?" GOTO 25[illegible]
300 PRINT AT 19,0;"[illegible]" AT 20,0;"[illegible]
[illegible]" AT 21,0;"(MAYO DE
1985"
301 PRINT AT 14,0;"TOQUE CUALQU
IER TECLA Y EMPIEZA"
302 IF INKEY$="" THEN GOTO 302
303 GOTO 1
```

Pantalla

```
N U M E R O S   M A G I C O S

*POR FAVOR PERMITAME CONOCERLO*.

ESCRIBA SU NOMBRE  HUGO

TAMBIEN SU APELLIDO  FISH

GRACIAS HUGO ....

FECHA  3  DE  JUNIO  DE  1955

*USTED NO ESPERA LO QUE A CONTI-
NUACION EXPERIMENTARA >>> )
```

Pantalla

```
BIEN AHORA LE MOSTRARE UN CONJUN
TO DE NUMEROS...

      3  5  7  9 11  1
     13 15 17 19 21 23
     25 27 29 31 33 35
     37 39 41 43 45 47
     49 51 53 55 57 59

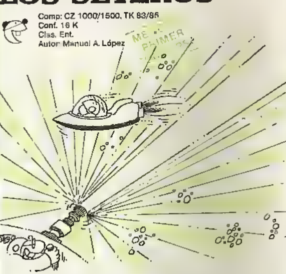
CONTESTEME (S/N) SU EDAD,
ESTA ENTRE ESTOS NUMEROS?...
```



CONCURSO TRIMESTRAL

LA INVASION DE LOS SETEROS

Comp: CZ 1000/1500, TK 83/85
Conf. 16 K
Clas. Ent.
Autor: Manuel A. López



LISTA DE VARIABLES

Z LOOP Controla el movimiento de la nave del encabezamiento.
Y LOOP Permite la entrada de las instrucciones.
X LOOP Tiña la pantalla hasta la línea 16 de negro.
W LOOP Tiñe la pantalla hasta la línea 21 de gris.
V LOOP Dibuja la cantidad de naves que quedan.
U LOOP Controla el efecto de la explosión.
T LOOP Controla el efecto de la explosión.
S LOOP Controla el efecto de la explosión.
R LOOP Tiñe la pantalla de negro.
Q LOOP Permite la entrada de la respuesta.
P LOOP Controla el movimiento de las naves del Fin del Juego.
O LOOP Tiña la pantalla de negro
N LOOP Controla el Scroll de las instrucciones.
MAX Puntaje máximo
PUN Puntaje
CON Controla la cantidad de naves que cruzaron la pantalla.
DISP Controla la cantidad de disparos efectuados por nave.
A Controla el movimiento en fila de la nave
B Controla el movimiento en columna de la nave.
C Controla el movimiento aleatorio de la nave.
D Control ascendente del disparo.
E Controla la diagonal del disparo 1.
F Controla la diagonel del disparo 2.
F Controla la diagonal del disparo 2.

Pantalla



G Controla la diagonal del disparo 2.

H Controla la diagonal del disparo 3.

TIPO DE PROGRAMA: Juego
CAPACIDAD DE MEMORIA: 4,4 K
MAQUINAS EN QUE PUEDE SER CORRIDO: TK 85 y similares cuya configuración mínima de memoria sea de 16 K
INSTRUCCIONES: Las mismas se encuentran dentro del programa.
TIPO DE CINTA: El programa fue grabado con una TK 85 en una cinta del tipo NORMAL sin ningún tipo de ecualización especial
NUMERO DEL CUENTAVUELTAS: 000

COMENTARIO DE LINEAS

1-85 Arman la presentación del juego

1-85 Arman la presentación del juego

90-160 Generan el movimiento de la nave en la presentación y permiten la entrada de las instrucciones pulsando el disparador (0).

170-330 Arman la pantalla del juego

340-510 Generan el movimiento aleatorio de la nave y la imprime (350-390 Controlan e imprimen la cantidad de naves que restan cruzar la pantalla)

520-540 Indican si se efectua un disparo y con cual base.

550 Si no se efectua un disparo la nave sigue avanzando.

560-690 Controlan el disparo de la base "1" (Subrutina)

700-870 Controlan el disparo de la base "2" (Subrutina)

880-1010 controlan el disparo de la base "3" (Subrutina)

1020-1170 Subrutina de disparo acertada de cualquier base.

1180-1285 Arman la presentación del fin del juego.

1290-1370 Generan el movimiento de las naves en la presentación del fin del juego y permiten la entrada de la continuación o no del mismo.

1380-1390 Grabación y corrida del programa

1500-1570 Menú del programa

1580 Permite el comienzo del juego

1590-1620 Scroll de la pantalla y comienzo del juego





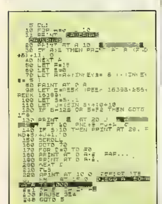
CONCURSO TRIMESTRAL

CAVERNAS

COMP: CZ-1000/TK-83/TK-85
CONF: 16K
CLAS: ENT
AUTOR: Miguel Angel García



Este programa nos desafía a recorrer la mayor cantidad de kilómetros posibles dentro de una caverna. Pero cuidado, hay peligrosos pozos trampa en los que podemos caer.



Pantalla



SISTEMAS

COMP: CZ-1000/TK-83/85
CONF: 2 K
CLAS: ENT
AUTOR: Diego Bonanata



Este programa resuelve sistemas de ecuaciones. Se utiliza el método de Crámer. La subrutina dispuesta, resuelve determinantes de n x n, según el método de Chío.

Las instrucciones son muy simples, sólo hay que responder a las preguntas y llenar los datos que nos pide la computadora.



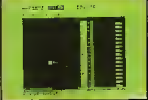
GUERRERO ESTELAR

COMP: CZ-1000/1500/TK-83/85
CONF: 16K
CLAS: ENT
AUTOR: Santiago Bustelo

Tenemos que destruir Ovnis que se agrupan. Se debe hacer un "agujero" entre los ovnis para que cuando despeguemos, no los choquemos. El resto de las instrucciones salen en pantalla.



Pantalla



```
  I REM PTA
  2 PRINT AT 9,0,"
                      ",AT I6,2
 "  I0 0  HECHO POR SANTIAGO BUST
ELO       DNI 14,863,354
          RIBERA 713 2º
 P         CAP 1085  BS,AS, JULI
  1985    PARA EL CONCURSO DE LA
REVISTA   "***", K 64  "***"
  3 PAUSE 45000
  4 LET MAYOR=0
  5 LET U$="NPO85."
  6 LET PU=0
  7 LET NA=3
  8 GOTO 2000
  9 PRINT AT 0,0,"PUNTOS  ";PU
RECORD  ",MAYOR;AT 20,0;"=.POR
U$
 I0 FOR O=2I TO 2I STEP -I
 II FOR A=2 TO I8
 I5 PRINT AT A,0,"            "A
A 0 "    "AT A,25,"=
 I6 NEXT A
 I7 PRINT AT I8,0,"
 
 I8 FOR U=I TO 5
 20 LET MISI=INT (RND*I6+2)
 2I LET MIS2=INT (RND*I6+2)
 22 LET MIS3=INT (RND*I6+2)
 27 LET MIS=AL
 25 IF INKEY$="0" THEN GOTO I40
 30 FOR U=I5 TO 5 STEP -I
 35 PRINT AT AL,8,"  " AT AL+I
,"  "AT AL-I,8,"  "
 36 PRINT AT MISI,"  "AT MIS
2,U,"  "AT MIS3,U,"  "AT MIS-
,"  "
 37 IF MISI=AL OR MIS2=PL AND U
=I0 THEN GOTO I00
 38 IF MIS3=AL OR MIS4=AL AND U
=I0 THEN GOTO I00
 40 LET PL=PL+(INKEY$="E" AND A
L<2I)-(INKEY$="7" AND PL>2)
 45 NEXT U
 46 PRINT AT MISI,"  ",AT MIS2
"  "AT MIS3,U,"  "AT MIS4,U,
"  "
 47 NEXT U
 50 NEXT O
 53 FOR P=6 TO 2E
 54 LET PL=AL+(INKEY$="E")-(INK
EY$="7")
 55 PRINT AT PL,A,"  "
 56 PRINT AT PL,(A+I
 58 LET P=PEEK (PEEK I6398+256*
PEEK I6399
 59 IF P=CODE "  "OR P=CODE "  "
THEN GOTO I50
 60 NEXT A
 62 CLS
 63 FOR P=I TO 30
 64 PRINT AT I0,I0,"BRAVO" AT I
0,I0,"BRAVO"
 65 NEXT A
 67 PAUSE 454
 68 LET P=P+I0
 70 GOTO 22I8
I00 CLS
II0 PRINT "UD.PERDIO UNA NAVE"
II2 LET NP=NP-I
II3 PAUSE 4800
II4 IF NP >0 THEN GOTO 20I8
II6 IF P>MAYOR THEN GOTO I20
II8 CLS
I20 PRINT "UD.PERDIO TODAS SUS
NAVES"
I22 LET NP=3
I23 PAUSE 4800
I24 CLS
I25 GOTO 206
I30 PRINT "BATIO EL RECORD EL A
ECORD"
I32 INPUT U$
I33 LET MAYOR=PU
I34 GOTO 8
I40 FOR A=5 TO 20
I42 PRINT AT AL,P
I45 LET P=PEEK (PEEK I6398+256*
PEEK I6399)
I46 IF P=CODE "  "OR P=CODE "  "
THEN GOTO I55
I48 PRINT AT AL,P,"  "AT AL,A
"  "
I49 NEXT P
I50 GOTO 35
I55 IF P=CODE "  "THEN LET PU=P
U+I0
I56 IF P=CODE "  "THEN LET PU=P
U+5
I57 PRINT AT C,E,"PUNTOS  "PU
I58 PRINT AT AL,P,"  "
I60 GOTO 35
2000 CLS
2002 PRINT AT I0,5,"GUERRERO EST
ELAR",AT II,5,"
    ",AT I5,0,"DESEA INSTRUCCION
ES?(S/N)"
2004 INPUT P$
2005 IF P$="S" THEN GOTO 2050
2007 IF P$="N" THEN GOTO 20I0
2008 GOTO 2004
2009 CLS
20I0 CLS
20I2 FOR A=I0 TO I5
20I4 PRINT AT A,I2,"          "
20I5 NEXT P
20I7 FOR A=I4 TO I4-(NP-I
20I8 PRINT AT II,A,"  "
2020 NEXT P
2022 CLS
2024 FOR A=I TO I8
2025 PRINT AT A,0,"
 
2026 NEXT A
2030 LET AL=I2
2040 GOTO 9
2050 CLS
2052 PRINT "EL OBJETIVO ES SOBR
EVIVIR: DESTRUIR OVNIS IGNOR
ANDO LAS NAVES QUE ESTAN MAS CE
RCA DE UD.DEBE HACER UN AGUJE
RO ENTRE LOS OVNIS."
2054 PRINT "PERO SUS PROPIOS MIS
ILES LO PUEDEN HACER EXPLOTAR,(
CONSEJO TIRE UN MISIL Y APARTE)"
2055 PRINT "ACENAS UD.TIENE 2 PR
OBLEMAS:  SI SE MUEVE RECIEN A
L SEO, DE A-PARECER EXPLOTARA,
          Y LOS LASERS NO ANDA
N HASTA QUE NO ANDEN LOS MISILES
 CONTRARIOS."
2060 PRINT AT 20,0,"APRIETE ENT
ER P/ SEGUIR."
2068 IF INKEY$="" THEN GOTO 20"
E
2074 GOTO 2060
2075 CLS
2077 PRINT "PUNTAJES--   AT 5,0
 PARTE BLANCA DEL OVNI 5 PUNTOS
 PARTE GRIS DEL OVNI I0 PUNTOS
2078 PRINT "ESTOS SON LOS CONTRO
LES"          6 P  IR A ABAJO
              7 P  IR A ARRIBA
              0 P  DISPARAR

              PULSE  ENTER  P/JUG
AR."
2085 IF INKEY$="" THEN GOTO 20I
E
2080 GOTO 2085
3I00 RUN
3000 SAVE "ATA"
3I00 GOTO I
```



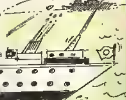
CONCURSO TRIMESTRAL

BATALLA NAVAL



COMP: CZ-2000/TS-2068
CONF: 48 K
CLAS: ENT
AUTOR: Oscar Fernando Montaña

Es una versión del popularísimo juego en el que hay que hundir al enemigo. Participan dos jugadores. Las instrucciones son claras y aparecen en pantalla.



Pantalla

Pantalla

```
2990 REM -CONTADOR DE HUNDIDOS-
3000 IF [illegible] THEN GO TO 3030
3010 LET [illegible] IF [illegible] THEN
[illegible]
3050 RETURN
[illegible] REM [illegible] JUEGO-
3500 LET [illegible] PRINT AT [illegible] FLASH 1 [illegible]
[illegible]
3530 FOR [illegible] TO 10 [illegible]
[illegible]
3560 PRINT AT 21,0 [illegible]
[illegible]
3590 IF [illegible] THEN GO TO [illegible]
3600 GO TO 10000
3990 REM - INSTRUCCIONES -
4000 GO SUB 4500
4010 PRINT AT [illegible]
[illegible]
4050 PRINT AT [illegible]
[illegible]
4100 PRINT AT [illegible]
[illegible]
4500 STOP
4510 CLS : PAPER 6: INK 5: PRINT
AT 0,8; [illegible] INSTRUCCIONE
[illegible]
6990 REM - GRAFICOS -
7000 PRINT AT [illegible] FLASH 1
[illegible] FLASH 0 [illegible]
[illegible] ESPERE
[illegible] 255 128 [illegible] 155 [illegible]
[illegible] 255 128 [illegible]
[illegible] 255 168 [illegible]
[illegible] 255 [illegible]
[illegible] 255 [illegible] 255 255 [illegible]
[illegible] 189 [illegible] 189 [illegible]
[illegible] 255 255 [illegible]
[illegible] 255 [illegible]
[illegible] 255 [illegible] 165 [illegible]
[illegible] 255 129 129 [illegible]
[illegible] 255 [illegible] 135 [illegible]
[illegible] TO 151
[illegible] TO 7
[illegible] USR CHR$ [illegible]
[illegible] 10000 [illegible] +5
[illegible] NEXT [illegible] NEXT I
[illegible] RETURN
7990 REM -MATRIZ [illegible]-
8000 FOR [illegible] TO 10
8010 [illegible]
8020 [illegible]
8030 NEXT [illegible]
8040 RETURN
```

SELECCION
II CONCURSO TRIMESTRAL
ENTREGAMOS 50 PREMIOS

Premios: RESET

SORTERAS, CARLOS G. CAPITAL
D'ANGELO, EDUARDO CAPITAL
CORONADO, FEDERICO CAPITAL

Premios: AURICULAR STEREO

BACA JUBAL, G.S.M.TUCUMAN
MONTAÑA, OSCAR F CORDOBA

Premios: CALCULADORA AURORA

COPELLO, MANUEL R. CORDOBA
GASPAROTTO, LUIS ADROGUE

Premios: CASETTE A/ELECCION

JUBAL, A.S.M. TUCUMAN
JUBAL, SEBASTIAN TUCUMAN
FRENKEL, DIEGO S FERNANDO

Premios: BECAS CURSO BASIC

IGLESIAS, DANIEL CAPITAL
OLIVERAS, ALBERTO SANTA FE
ASAD, SERGIO ROSARIO
CHILLA, JUAN L.A. CAPITAL
DUARTE, JUAN JOSE CAPITAL
ALONSO, MIGUEL CAPITAL

PERSAK, GUSTAVO N. CAPITAL
SANCHEZ, HECTOR NEUQUEN
CARZO, LUIS CAPITAL
DEL LAGO, EDUARDO CAPITAL

Premios: CASETTE

PALUMBO, JAVIER LINCOLN
GRAZIOLA, H A BARILOCHE
RODRIGUEZ L PUNTA ALTA
CUESTA, DANIEL TANDIL
ENDEMAÑO, HECTOR QUILMES
MARQUEZ, MARCELO M PLATA
RUBIO, MARIO R. ROSARIO
RUTMAN, GUILLERMO ROSARIO
GIODA, MARCELO RIO IV
PAGNI, JUAN E CORDOBA
MARTINA, ERNESTO RESISTEN
AGOTE, TOMAS CAPITAL
RABATIN, GABRIEL CAPITAL
LUNA, OSCAR CAPITAL
ARENAS, ANTONIO G. CASEROS
BETTI, ELIO A. MENDOZA
LUPARIA, CARLOS H. CAPITAL

MARCHESE MARCELO CAPITAL
SICCARDI DIEGO CAPITAL
MONTAÑA OSCAR CORDOBA
SILVA CARLOS SAN MARTIN
GUTIERREZ SERGIO PAMPA
VERA RAUL SGO. DEL ESTERO
PELLACANI ESTEBAN A.
MATOSO, GUSTAVO A CAPITAL
DONATO, JOSE CAPITAL
MORENO, JULIO CAPITAL
SARMIENTO, HERNAN C GOMEZ
VALLOTA, MARCELO CAPITAL
MATEO, ALEJANDRO S. ROSARIO

Los premios podrán retirarse en la Administración de K 64, Cerrito 1320, piso 1º, Capital, con documentos de identidad, en el horario: 10 a 12 y 15 a 17 horas.
Quienes viven en el interior del país, pueden solicitar que se les remitan los premios por correo

CUANDO LA MEMORIA NO ALCANZA

En primer lugar debo felicitarlos por tan muy buena labor que están llevando a cabo.
Les escribo además estas humildes líneas dado que poseo una TK-83 (sin expansión) y he querido hacer el programa de radiotelegrafía publicado en los números 2 y 3, pero me encuentro en que una sola línea REM la máquina no me acepta los 1470 caracteres, por ello he probado elaborando una línea de 100 caracteres y editándolo 14 veces, pero al cargar los primeros 100 POKES, deja de tomar el código de máquina. Les agradeceré que me informen una solución.

Graciela Casas
Mar del Plata

P.D.: Tampoco pude utilizar el generador de REM del programa simulador de vuelo, porque no me alcanza la memoria disponible

K64

Mirá Graciela tenemos un kilo de cartas que nos preguntan prácticamente lo mismo que vos, así que atención a todos.



Debo remitirme a las respuestas dadas por el propio autor del programa en el Nº 5, sección correo. Nadie mejor que él para responder estas consultas.
Por otra parte, es muy lógico que el generador de REM no te funcione: es para 16 K.

ROM

Deseo que detallen el funcionamiento del programa PANT MOTOR; publicado en el Nº 4 y me comuniquen que significa eso de "permite cargar imágenes que han sido grabadas en el cassette..."
En segundo lugar, quiero saber porqué el ejecutar dicho programa con una TK-85 se bloquea la pantalla con una llovizna y no responde a ninguna tecla, incluso "BREAK", y para recuperar la imagen debo apagar el transformador.
Solicito que me indiquen porque no funciona el programa simulador de vuelo, en la generación de los 2926 caracteres de REM.

Gustavo Persak
Ceballito

K 64

Gustavo, la aclaración del programa Pant, te la quedamos debiendo por ahora.
Respecto al generador de REM, fijarse en la nueva sección DEBUGGINS, para errores y otros piffies.
Por otro lado la ROM de un TK-85 es un poquitito diferente a la ROM de SINCLAIR; dado que el TK-85 tiene algunas funciones de más y esas también forman parte de la ROM. Es posible que las direcciones que ocupan estas "extensiones", al no estar disponibles para la memoria RAM, hagan que el programa se bloquee.

Queremos reiterar nuestra intención de contestar a todas las cartas que nos llegan.
Sin embargo, nuestro correo se está transformando en una "hot line" que no sabe de pausas. Por ello les pedimos un paquito de paciencia a aquellos "impacientes" y les contamos que nuestros procesadores de texto (hace rato abandonamos la OLIVETTI), de la sección consultas están trabajando a full, dedicados a sus cartas.
Hasta aquí todas son flores, (o casi), pero aquí viene el "de paso cañazo...", ¿o pensaban sacarla barata?. Por favor, sería muy provechoso para todos dedicarle algunos segundos más de lo habitual cuando escriban sus cartas, porque la verdad, tenemos algunas que son verdaderas obras maestras de la simbología egipcia, y por la redacción egiptólogos todavía no hay.
Por eso les pedimos que escriban a máquina y si no disponen de tan prolijo elemento, escriban con letra de imprenta que sea lo más clara posible!. De lo contrario, quedarán archivadas para futuras investigaciones.
Bueno, hasta aquí hemos planteado nuestras consultas hacia ustedes lectores. Sólo nos resta felicitarlos por el nivel que estan alcanzando como usuarios y ratificar nuestro empeño en este correo que no nos da tregua.
_K-64_

VOZ Y/O SONIDO

Soy usuario de una TS-1000 y tengo información acerca de la posibilidad de hacer voz y/o sonido con esta máquina sin necesidad de reforma alguna ni agregado de periférico alguno.
Se me ocurre que dicha información no debe ser muy descabellada ni distante de la realidad.
¿Es esto posible?

Leonardo Szapiro
Bs. As.

K 64

Cuidado con el y/o. Porque hacer sonido es una cosa, y hacer que ese sonido sea además inteligible para el oído humano es otra, y muy distinta. Pero en resúmen sí, es posible generar sonidos sin periféricos, sólo por software, aunque ese sonido debés tomarlo de la salida MIC, lógicamente. Con respecto a la generación de voz, ni noticias

EL ORGANO NO FUNCIONA

Los felicito por la calidad de revista que editan. Los programas son buenísimos. Necesito que me solucionen un problema en la revista número 2, en la página 33 publicaron el programa ORGANO para el Spectrum. Lamento informarles que no funciona. Hay variables que no están definidas

Andres Ravizzini
Rosario

K64

Estimado Andrés, pongemos en práctica nues-

tros conocimientos del BASIC. La respuesta a tu carta es.
GO TO debugging, Nº 6
Y gracias por las felicitaciones.

SR. DIRECTOR DE K-64:

Soy poseedor de una Commodore 64 y entre los artículos que aparecen en la revista me interesó el programa para duplicar software, pues justamente como ustedes lo mencionan en su artículo, al colocar un diskette que yo había comprado, un compañero mio lo borró y perdí todos los programas que me habían grabado y estos tenían protección. Por lo tanto me sería de mucha utilidad saber el la revista va a publicar un programa similar para Commodore 64

Roberto Junco
Capital

K64

Lamentamos mucho la pérdida y esperamos que sigan siendo amigos
Ahora un poco más en serio, cuando la información cae en nuestras manos, va directo a las páginas de nuestra revista, para que los lectores la exploten al máximo. En función de esto, te decimos que ni bien tengamos algo que te resulte útil, lo vas a ver publicado.

INTERCAMBIO DE PROGRAMAS

Les escribo para felicitarlos por su revista. Me pareció fascinante el número 3, muy interesante sus comentarios sobre EXPOUSUARIA '85 y todo el contenido en general.
Me gustaría saber si se podrá adaptar el programa Simulador de Vuelo, del Nº 3 para una TI 99/4A
Me gustaría comunicarme con chicos que tengan programas para jugar con la TI99/4A.

Hernán Sarmiento
Cañada de Gómez

K64

Gracias por tus felicitaciones.
En cuanto a adaptar el simulador para tu TI, tal y como está en el listado publicado, te voy a citar una palabra que alguna vez dijo nuestro quando "Lole" Reutemann. Es difícil, muy difícil...
Bueno el pedido de comunicación está dado por Hernán hacia todos los usuarios de TI 99/4A. Para que se comuniquen, les damos la dirección: Hernan Enrique Sarmiento (13 años)
Ocampo 1252
(2500) Cañada de Gómez

DESARROLLO NACIONAL

Soy un entusiasta lector de su revista desde el primer número y uno de los tantos "teóricos" de la computación, que por uno u otro motivo todavia no han accedido a una micro. No obstante pienso adquirir una CBM-64. Sobre este particular desearía una respuesta sobre de Drean, al respecto de cuando sacan la versión nacional de la "CBM-64"
Las máquinas que se consiguen actualmente, se compran casi a ciegas, dependiendo su test de buen funcionamiento, de la demostración que hace en el momento el vendedor.
Desearia sugerir la introducción de una sección de la revista destinada a niños y adolescentes de distintas edades y pensada en función de las características culturales y sociales de nuestro país, evitando les meras adaptaciones de productos importados.
Por favor, sigan el camino elegido

Alfredo Luis Berte
Lanús

K64

Lo felicitamos por su decisión de "comprarse una micro".
Esperamos obtener una respuesta de Drean. Lo que sabemos por ahora, es que comenzaron comercializando sólo el software, antes de lanzar el equipo.
Respecto de la compra de una máquina, el hecho de que "salga andando de primera", es un síntoma bastante bueno.
La nueva sección que nos sugiere, es una idea excelente, pero para eso los **"si tuviésemos"** *son demasiados.*
"si tuviésemos la tecnología..."
"si tuviésemos desarrollo propio y nacional de software..."
"si tuviésemos una idiosincracia abierta a la computación..."
"si tuviésemos usuarios de micros de todas las edades..."
En fin la lista continúa y es bastante larga, tal vez en un futuro muy próximo, estemos en condiciones de dar fundamento a esa seccion. Así lo esperamos, y nuestro esfuerzo apunta a ello.

INTERFACES

Espero, proximamente comprarme una TK-90 X y quisiera saber si existe en venta en Argentina algún tipo de drive o microdrive para diche computadora y si necesita alguna interface u otra cosa para hacerla funcionar.

Eduardo CZAR
Cap

K64

Lamentablemente, no hay nada de eso. Y por otra parte es normal y necesario que cualquier sistema de este tipo se apoye en interfaces para "hacerla funcionar". Lo que ocurre es que todo este tipo de sistemas deben incluir nuevos comandos y órdenes que se agregan como extensión a los comandos de la máquina y éso entra via interface. Generalmente los sistemas existentes en Europa se venden todo junto, esto es interface mas drive etc.

COLABORACIONES

Ha leído con sumo interés los dos últimos números de vuestra revista K64 considerando que la misma es la de mejor factura entre las publicadas en el país.
Es esta una de las razones por las cuales me ofrezco como colaborador de vuestra revista, con artículos que pueden ser del interés de vuestros lectores.
Poseo una computadora TS 2068 con impresora la cual funciona también como SPECTRUM con Interface I y Microdrive.

Ing. Gonzalo Estivariz
Concordia
Entre Ríos

K64

Nos interesa recibir colaboraciones de nuestros lectores.
Las notas ofrecidas por usted están dentro de nuestros temas, por lo cual le propuesmos que nos envie el material para que sea considerado por nuestro comité editorial
Por notas publicadas los autores recibirán una orden de compra de elementos de computación, por un monto a determinar.

FELICITACION

Deseo felicitarles por su revista, no sólo por su contenido sino por el entusiasmo con que la editan.
Soy un usuario de una computadora tipo industrial, y no hace mucho adquirí un microcomputador TS 1000 y gracias a su revista uno no pierde el entusiasmo de usar ese pequeño aparatito que nos enseña y entretiene por horas.
Atentamente.
Claudio Aranda

CORREO • CONSULTAS

BATALLA NAVAL

En el diario TIEMPO ARGENTINO ha llegado a mi poder el programa denominado "Batalla Naval", entre otros. El problema está en que he intentado correr dicho programa en mi Sinclair 1500 y éste no responde como debiera.

Fernando D. Bertone
Tandil - Buenos Aires

K-64

En el número 2 de la revista K64 salió el programa "Batalla Naval" correctamente, pero si usted no la posee, le informamos que salió mal impresa en el diario la línea Nº 100. La correcta es: 100 INPUT T

RTTY y SSTV

Tengo el agrado de dirigirme a usted a fin de felicitarlo por la prestigiosa publicación que dirige que, evidentemente día a día tiene más lectores ya que se hace difícil, a veces, conseguirla a veinticuatro horas de lanzada al mercado.

Por otra parte, y abusando de su amabilidad le solicito si tiene información respecto de algún programa como para RTTY y SSTV (barrido lento), así como han publicado el de CW (telegrafía). El motivo de mi pedido es que poseo una Sinclair 1500 y, como también la utilizo como radioaficionado, quisiera aprovecharla en los modos arriba mencionados.

Lamentablemente, hay poca información al respecto y su forma de publicación es muy clara, so-

bre todo para quienes recién se inician, como es mi caso (son bastante contemplativos).
En lo que respecta al programa de CW (telegrafía), es sensacional y brillante el trabajo del ingeniero Pedro E. Colla, que logró lo que muchos estaban esperando y nunca se había podido concretar, por falta de conocimientos o porque tal vez, el que lo conocía, era "un poquito reservado", caso totalmente opuesto al ingeniero Colla. El demostró ser un excelente profesional y, a la vez, gran colaborador del resto de los Radioaficionados que tuvieran interés por instruirse en el tema.
Luis Agostini
LU2ALH

K-64

La Sinclair 1500 posee unas limitaciones que hacen imposible decodificar RTTY o SSTV solamente por Software. No es así en cambio, el decodificar RTTY con la ZX Spectrum o la TS2068. Esperamos publicarlo pronto.

SORTEO-ENCUESTA K64

LLENE ESTE CUPON Y PARTICIPE DEL SORTEO MENSUAL



50 PREMIOS:
40 CASSETTES Y 10 BECAS PARA CURSOS BASIC

NOMBRE: .... EDAD: .... OCUPACION: .... TEL: ....

DIRECCION: .... C.P.: .... LOCALIDAD: .... PCIA: ....

ENCUESTA

COMPUTADORA: ☐ CZ 1000 ☐ CZ 1500 ☐ CZ 2000 ☐ TIMEX 2068 ☐ TK 83 ☐ TK 85 ☐ TK 90
☐ C-16 ☐ C-64 ☐ TI 99/4A ☐ Otras .... ☐ NO TENGO AUN

ME GUSTARIA VER:

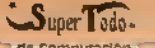
MAS	IGUAL	MENOS	
☐	☐	☐	PROGRAMAS DE APLICACION ESPECIFICA
☐	☐	☐	PROGRAMAS EN BASIC
☐	☐	☐	PROGRAMAS EN LOGO
☐	☐	☐	PROGRAMAS EN LENGUAJE DE MAQUINA
☐	☐	☐	PROGRAMAS EN OTROS LENGUAJES
☐	☐	☐	ANALISIS DETALLADOS DE LOS PROGRAMAS
☐	☐	☐	NOTAS PARA BEGGINERS

MAS	IGUAL	MENOS	
☐	☐	☐	JUEGOS
☐	☐	☐	CALIFICACION DESCRIPTIVA DE PROGRAMAS DE JUEGO
☐	☐	☐	PROGRAMAS DE APLICACIONES COMERCIALES
☐	☐	☐	PROGRAMAS EDUCATIVOS
☐	☐	☐	HARDWARE

QUE ES LO QUE MAS TE GUSTA DE K64?

QUE ES LO QUE MENOS TE GUSTA?

Enviarlo a: K64 Computación Para Todos - Cerrito 1320 1° (1010) Buenos Aires, Rep. Argentina


el
SuperTodo
de Computación